

N° d'ordre : 3760

THÈSE

Présentée par

FARID BOUDJEMAÏ

Pour obtenir le titre de

DOCTEUR

de l'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Discipline : AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

RECONSTRUCTION DE SURFACES D'OBJETS 3D A PARTIR DE NUAGES DE POINTS PAR RESEAUX DE NEURONES 3D-SOM

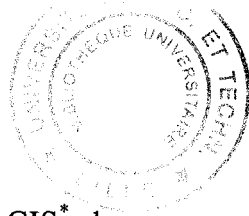
Soutenue le 08 février 2006 devant la Commission d'Examen :

OLIVIER COLOT	Président	Professeur à l'USTL
DENIS HAMAD	Rapporteur	Professeur à l'ULCO
FREDERIC MERIENNE	Rapporteur	Professeur à l'Institut Image - ENSAM
LESZEK LUCHOWSKI	Examineur	Chercheur à l'IITiS PAN Pologne
JACK-GERARD POSTAIRE	Co-Directeur de thèse	Professeur à l'USTL
PHILIPPE BIELA ENBERG	Co-Directeur de thèse	Enseignant-Chercheur à HEI

Thèse préparée au LAGIS UMR 8146 et à HEI - ERASM



Remerciements



Ce travail de recherche à été réalisé sous la responsabilité du laboratoire LAGIS* de l'Université des Sciences et Technologies de Lille, il représente l'aboutissement de travaux que j'ai mené dans le cadre de mes activités au sein d'HEI* et d'ERASM* à Lille.

Je remercie à ce titre Monsieur Olivier COLOT, Professeur à l'Université des Sciences et Technologies de Lille pour avoir accepté la présidence du jury de cette thèse.

Je tiens également à adresser mes plus vifs remerciements au Professeur Jack-Gérard POSTAIRE directeur de cette thèse de m'avoir accueilli au sein de son équipe, du soutien qu'il m'a apporté tout au long de ces années et de m'avoir guidé et conseillé dans mes recherches.

J'exprime toute ma gratitude à Philippe BIELA ENBERG, Chef du Département Automatique à HEI, pour avoir su écouter et répondre à mes innombrables questions, et pour son aide précieuse durant ces années.

Je remercie également les professeurs Denis HAMAD, de l'Université du Littoral Côte d'Opale, et Frédéric MERIENNE, de l'Institut Image – ENSAM de Cluny, pour m'avoir fait l'honneur de juger mon travail.

Mes remerciements s'adressent à Monsieur Leszek LUCHOWSKI de Institut d'Informatique Théorique et Appliquée de l'Académie Polonaise des Sciences IITiS PAN pour avoir accepté d'examiner ce travail.

Je remercie la Région Nord Pas-de-calais et la fondation Norbert Ségard pour avoir financé ce travail de recherche.

Enfin je remercierai tous mes amis et collègues de travail d'HEI et du LAGIS pour l'aide qu'ils ont pu m'apporter au cours de ces quelques années en me témoignant de leur sympathique et amical soutien.

Pour terminer, je ne peux oublier d'exprimer ma reconnaissance envers mes parents, ma grand-mère, mon frère, mes sœurs et tout le reste de la famille, sans oublier les amis, pour leur présence à mes côtés et pour leurs encouragements.

*LAGIS : Laboratoire d'Automatique, Génie Informatique et Signal.

*HEI : Hautes Etudes d'Ingénieur

*ERASM : Equipe de Recherche en Automatique des Systèmes et Microsystèmes.

A mes parents.

Table des matières

Introduction générale 9

1. La reconstruction de surfaces à partir d'un nuage de points non-organisés..... 9

2. Notre contribution..... 10

3. Organisation du manuscrit..... 10

Reconstruction de surfaces à partir de nuages de points non-organisés 12

1. Introduction 12

2. Acquisition du nuage de points..... 14

2.1. La numérisation par contact..... 15

2.2. La triangulation..... 15

3.3. La télémétrie 16

2.4. Les méthodes optiques passives..... 16

3. Approches basées sur le diagramme de Voronoï..... 16

3.1. Le Crust 18

3.2. Le Cocone..... 20

3.3. Le Powercrust 20

4. Approches implicites..... 21

5. Approches mixtes..... 25

5.1. Méthode de Hoppe 25

5.2. Méthode mixte de Boissonnat..... 26

6. Approches dynamiques 27

7. Approches neuronales 29

7.1 Méthodes utilisant les cartes auto-organisatrices de Kohonen 30

7.2 Méthodes utilisant les « Growing Cell Structures »..... 31

8. Conclusion..... 31

Les cartes auto-organisatrices de Kohonen 32

1. Introduction 32

2. Fondements biologiques..... 32

3. Réseaux compétitifs 37

3.2. Stratégie d'apprentissage compétitif (« Winner Takes Most ») 39

3.3. Extensions et développement des réseaux compétitifs..... 40

4. Carte auto-organisatrice de Kohonen..... 42

4.1. Architecture et topologie de la carte de Kohonen..... 43

4.2. Principe général de l'algorithme de Kohonen 46

4.3. Algorithme de Kohonen 50

4.4. Exemples d'application 51

5. Convergence de l'algorithme..... 55

6. Conclusion..... 56

Les réseaux 3D-SOM 57

1. Introduction 57

2. Architecture des réseaux 3D-SOM 58

2.1. Introduction..... 58

2.2. Généralisation de l'architecture des cartes auto-organisatrices 60

2.3. Subdivision totale..... 61

3. Apprentissage compétitif..... 64

3.1. Apprentissage compétitif simple	64
3.2. Apprentissage compétitif avec répulsion	67
4. Initialisation du réseau 3D-SOM.....	72
4.1. Initialisation aléatoire.....	72
4.1. Initialisation « pré-établie ».....	73
5. Etude de l'effet des paramètres d'adaptation sur les 3D-SOM.....	75
5.1. Sensibilité de l'apprentissage par rapport au coefficient d'apprentissage $\alpha(t)$	75
5.2. Sensibilité de l'apprentissage par rapport au niveau du voisinage d'adaptation $N_c(t)$	76
6. Chaîne d'acquisition/ reconstruction par 3D-SOM.....	78
6.1. Matériel utilisé pour l'acquisition de points.....	78
6.2. Mise en œuvre du réseau 3D-SOM et résultats	80
7. Conclusion	82
Améliorations du processus d'adaptation des 3D-SOM	83
1. Introduction.....	83
2. Prise en compte de la densité de points dans le processus d'adaptation	84
3. Défauts d'adaptation.....	87
4. Réadaptation.....	89
5. Subdivision/Adaptation locale	91
5.1. Identification des triangles non-représentatifs	91
5.2. Subdivision locale.....	92
5.3. Apprentissage des zones subdivisées et des zones non subdivisées.....	94
6. Prise en compte d'un critère d'orientation de la normale à la surface pour chaque triangle	97
6.1. Intérêt du critère d'orientation.....	97
6.2. Implantation du critère d'orientation dans l'algorithme des 3D-SOM.....	99
7. Exemples de reconstructions de surfaces par 3D-SOM	101
8. Conclusion	105
Conclusion générale	106
1. Conclusion	106
2. Perspectives.....	106
Références personnelles	108
Bibliographie	109

Introduction générale

1. La reconstruction de surfaces à partir d'un nuage de points non-organisés

Les travaux concernant la reconstruction de surfaces à partir de nuages de points non-organisés ont probablement été abordés pour la première fois dans les années 80, en particulier par Boissonnat [Boi84]. Dans ce contexte, cette reconstruction a pour but de générer un modèle numérique de la surface externe d'un objet physique à partir d'un nuage de points dont les coordonnées 3D sont saisies par un dispositif de mesure.

Une telle problématique se retrouve dans beaucoup de domaines, par exemple en vision par ordinateur, en reconnaissance de formes, en informatique graphique pour les jeux et les représentations artistiques et dans le « Reverse Engineering » (rétro-conception), qui consiste à créer le modèle numérique d'un objet en vue de le reproduire, d'en réaliser un prototype ou de fabriquer l'outillage pour le réaliser.

Un grand effort de recherche/développement a été porté vers ce secteur ces dernières années quand les dispositifs de numérisation de surfaces sont devenus suffisamment bon marché pour être mis à la disposition d'un large éventail d'utilisateurs. En effet, la solution à ce problème n'est pas triviale et beaucoup de méthodes ont été développées pour tenter d'y apporter une solution.

L'étape de départ consiste à décrire la surface d'un objet par un ensemble de points, cette opération est appelée numérisation. On obtient alors un fichier de points qu'il faut ensuite traiter afin d'obtenir un modèle numérique de l'objet. Les fichiers qui regroupent les mesures 3D obtenues sont appelés « nuages de points ». Il est, bien sûr, très important d'insister sur le fait que la précision du modèle numérique qui est généré ne sera jamais supérieure à la précision de mesure.

Dans le cadre de ce manuscrit, nous nous intéressons plus particulièrement à la reconstruction sous forme de maillage des nuages de points non-organisés. Le processus de reconstruction fait en sorte de représenter les points sous la forme d'un ensemble de triangles. Cet ensemble forme une surface polygonale, représentant l'objet, qu'on appelle maillage de la surface.

Il existe plusieurs méthodes de reconstruction de surfaces à partir de nuages de points non-organisés. Néanmoins, l'apport des réseaux de neurones à ce domaine est relativement récent

et s'annonce riche en résultats encore à venir. Dans ce cadre, nous introduisons une méthode basée sur une approche neuronale appelée 3D-SOM.

2. Notre contribution

Les travaux menés dans le cadre de cette thèse ont pour principal objectif de développer une architecture neuronale auto-adaptative et non supervisée dévolue à la reconstruction d'objets 3D. L'espace d'entrée du réseau de neurones est constitué du nuage de points 3D non-organisés acquis à la surface de l'objet à reconstruire. Nos travaux se sont orientés vers le développement d'une nouvelle architecture neuronale inspirée des cartes auto-organisatrices de Kohonen que nous avons appelée 3D-SOM. Après traitement du nuage de points, le réseau se présente sous la forme d'une architecture neuronale maillée ayant pris la forme de l'objet à reconstruire.

Nous avons adjoint au réseau 3D-SOM différentes lois d'apprentissage, permettant la mise en œuvre d'un processus d'auto-organisation efficace. Nous avons également introduit :

- un niveau d'adaptation modulable qui permet une évolution dynamique du voisinage d'apprentissage.
- une loi d'adaptation locale subordonnée pour des critères spécifiques à chaque neurone ou pour chaque triangle (coefficient d'apprentissage local, densité locale des points dans le nuage, orientations des normales aux triangles).
- un processus de subdivision d'architecture global ou local qui consiste à accroître le nombre de neurones (donc du nombre de mailles) composant le réseau aux endroits où le réseau rencontre des difficultés à s'adapter à la forme réelle de l'objet.

3. Organisation du manuscrit

Le premier chapitre présente un bref survol des différentes approches existantes pour la reconstruction de surfaces à partir de nuages de points non-organisés.

Dans le second chapitre nous présentons de manière détaillée les cartes auto-organisatrices de Kohonen en insistant sur les références biologiques, les lois d'apprentissage, les architectures et les applications. Ce chapitre nous permet aussi de mettre en évidence certaines limitations du modèle de la carte auto-organisatrice dans le domaine de la reconstruction de surface.

Ayant acquis les concepts de base et les principes essentiels des cartes auto-organisatrices dans le chapitre 2, nous abordons le chapitre 3 pour y présenter l'architecture et les spécificités comportementales des 3D-SOM, en présentant des exemples de reconstructions basés sur différentes stratégies d'adaptation et d'initialisation du réseau. Puis nous étudierons l'influence des paramètres du réseau sur son adaptation et nous finirons par un exemple illustrant la chaîne d'acquisition/reconstruction par 3D-SOM.

Enfin, dans le dernier chapitre nous proposons des solutions pour améliorer les performances des 3D-SOM et éliminer les défauts d'adaptation. Nous illustrons la qualité de la représentation d'objets 3D à l'aide des 3D-SOM par différents exemples de reconstruction.

Reconstruction de surfaces à partir de nuages de points non-organisés

1. Introduction

La reconstruction d'une surface à partir d'un nuage de points non-organisés peut être énoncée comme suit : nous disposons, en entrée du processus de reconstruction, d'un ensemble de points non-organisés acquis à la surface d'un objet et le but recherché est de produire en sortie une surface approchant au mieux la forme de la surface physique de l'objet échantillonné (cf. figure 1.1).

Le nuage de points représentant l'objet est généralement obtenu par numérisation manuelle ou par scanner. Chaque point peut être accompagné d'informations additionnelles, en général la normale à la surface en chaque point et l'information de couleur. Un point est représenté par ses 3 coordonnées spatiales.

La reconstruction de la surface représentée par le nuage de points est abordée de différentes manières selon la méthode utilisée. On peut classer les différentes approches selon la façon dont la surface est reconstruite, par approximation ou interpolation, et la façon dont est traité le nuage de points, de manière globale ou locale.

L'interpolation signifie que la surface reconstruite préserve le nuage de points original, les points échantillonnés sur la surface de l'objet feront partie de la surface reconstruite.

L'approximation signifie que la surface est très proche des points échantillonnés mais ne passe pas obligatoirement par eux.

Les méthodes d'approximation sont beaucoup plus robustes aux erreurs d'acquisition des données 3D que les méthodes d'interpolation qui sont très sensibles aux erreurs d'acquisition et aux bruits de mesure. Par analogie au domaine du traitement du signal, on appelle « bruit de mesure » le fait que les valeurs mesurées ne sont jamais exactes mais un peu au dessus ou en dessous de la valeur réelle.

Une autre classification possible consiste à différencier les approches globales des approches locales.

Dans les méthodes globales, tous les points du nuage sont employés pour définir la surface reconstruite.

Par contre, dans les méthodes locales, la reconstruction de la surface se fait de manière locale, par groupes de points, ou point par point et c'est la combinaison de ces reconstructions locales qui fournit une reconstruction globale de la surface.

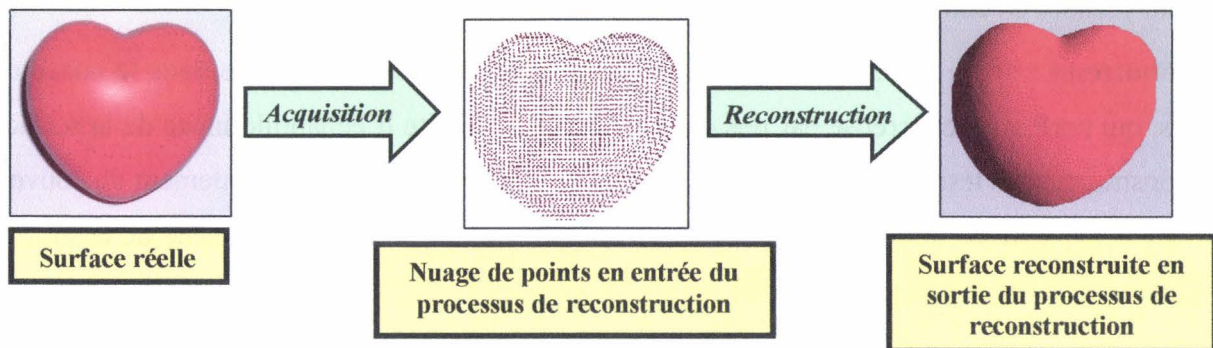


Figure 1.1 Exemple de reconstruction de la surface d'un objet à partir d'un nuage de points non-organisés acquis à sa surface. Modèle : Heart / Scanner : 3D Roland PIX-30 / Nombre de points : 3093 / Source du nuage de points : Département de mathématique, Université de Bologne, Italie.

Ces deux façons de classer les méthodes de reconstruction en approches globales ou locales ou en approches d'interpolation ou d'approximation, sont assez générales mais elles ne reflètent pas, de manière suffisamment discriminante, la multitude des approches existantes.

L'objet de ce chapitre n'est pas de présenter de façon exhaustive toutes les approches de reconstruction existantes mais nous essayerons de présenter de façon succincte certains algorithmes. Ainsi ce bref survol du sujet permet de montrer l'entendu du domaine et la multitude des méthodes existantes, tout en mettant en relief les lacunes et les avantages de chaque approche.

La classification des approches de reconstruction de surfaces peut être réalisée selon les différents algorithmes et méthodes qu'elles utilisent. Notre classification s'inspire de celles proposés par Varnuska [Var05] et Schall [Sch05]. Nous avons choisi de présenter les méthodes les plus récentes, sans oublier les méthodes qui sont à la base de la reconstruction de surfaces à partir de nuages de points telles qu'on les connaît aujourd'hui [Hop92].

D'une manière générale, on peut classer les méthodes de reconstruction à partir d'un nuage de points non-organisés en plusieurs grandes familles : les approches implicites [Tur02, Reu03], les approches fondées sur le diagramme de Voronoï [Ame98, 99, 01], les approches

mixtes qui combinent plusieurs méthodes [Hop92, Boi00], les approches dynamiques [Qin01] et enfin les approches neuronales [Baa94, Yu99, Iv03].

Par ailleurs, il existe des contraintes d'efficacité que les algorithmes de reconstruction doivent prendre en compte. Elle peuvent être énumérées comme suit :

La robustesse : degré de sensibilité de la méthode au bruit d'échantillonnage dans le nuage de points, qui parfois se concrétise par des conditions sur la densité d'échantillonnage de la surface à reconstruire, sans lesquelles la reconstruction serait incohérente. Il s'agit également de pouvoir évaluer la capacité de la méthode à prendre en compte certains détails de la forme de la surface originale (bords anguleux par exemple).

Information additionnelle : capacité de la méthode à reconstruire la surface sans avoir besoin d'informations additionnelles, telle que la normale à la surface pour chaque point d'entrée.

Ressource mémoire : le nombre de points en entrée du processus de reconstruction est un facteur très important en raison de la quantité de mémoire disponible sur les machines de calcul.

Vitesse de calcul : la reconstruction d'une surface à partir d'un nuage de points est habituellement un processus off-line, la rapidité d'obtention d'un résultat de reconstruction 3D est donc l'une des contraintes les moins critiques. Cependant cette contrainte dépend de l'application.

2. Acquisition du nuage de points

Pour réaliser la numérisation d'une surface afin d'obtenir un nuage de points 3D la représentant, différentes techniques sont disponibles.

Une proposition de classification des méthodes d'acquisition de nuages de points 3D a été proposée par Goulette [Gou00]. Elle consiste à distinguer quatre catégories : la numérisation par contact, la télémétrie, la triangulation et les méthodes optiques passives.

Ces différentes méthodes fournissent des nuages de points organisés ou non-organisés. Un nuage de points est dit « organisé » ou « structuré » s'il est possible de connaître les voisins immédiats de chaque point. Par opposition, un nuage de points est dit « non-organisé » ou « non-structuré », dans les cas suivants :

- La structure des points n'est pas connue.
- Les points ont été mesurés dans un ordre complètement aléatoire.

- Un traitement intermédiaire « a mélangé » les points de telle sorte que le fichier ne contient plus de relations de voisinage.

2.1. La numérisation par contact

Pour le contrôle dimensionnel de cotes 3D dans un cadre industriel (métrologie), la numérisation par contact a précédé les techniques sans contact. Les techniques par contact ont été exploitées, en premier, dans les années 80, dans l'industrie pour la numérisation 3D de pièces mécaniques en vue d'améliorer le processus de contrôle qualité.

Les techniques de numérisation par contact ou « palpage » consistent à mesurer la cote de points situés sur des surfaces, par contact avec ces points et repérage de la position d'un outil (stylet) de contact. Ce repérage peut se faire par procédé optique ou par mesure des positions angulaires d'un bras articulé au bout duquel se trouve le stylet. C'est une technique d'acquisition 3D simple, qui peut donner de très bonnes précisions. Son inconvénient est la lenteur d'acquisition, ce qui conduit souvent à limiter le nombre de points de mesures. Un autre inconvénient est que la surface doit être suffisamment rigide pour ne pas se déformer au moment du contact. Par ailleurs certaines zones peuvent ne pas être atteignables.

2.2. La triangulation

La triangulation consiste à projeter sur la surface à mesurer un point ou une ligne laser, puis à observer ce point ou cette ligne par un système optique décalé de l'émetteur (laser). Le système optique est généralement une caméra numérique. En connaissant la direction d'émission du rayon, la position observée par le récepteur (système optique) et les positions relatives de l'émetteur et du récepteur, on peut calculer la distance du point observé à l'émetteur, puis sa position tridimensionnelle dans l'espace d'observation. L'inconvénient principal, des méthodes de triangulation, réside dans le temps d'acquisition des points et dans le traitement des données pour éliminer les points n'appartenant pas à l'objet à numériser. Les systèmes les plus répandus de triangulation active sont ceux qui utilisent un laser (Scanners 3D). Pour l'ensemble des systèmes de triangulation, il est nécessaire de déplacer les têtes de mesure caméras-laser, pour réaliser une numérisation complète de la scène (stéréovision active).

3.3. La télémétrie

La télémétrie consiste à mesurer la distance entre l'objet et l'émetteur-récepteur dans la direction d'émission-réception d'une onde. La technique principale exploite le temps de vol pour laquelle le temps d'émission-réception est mesuré en connaissant la vitesse de propagation de l'onde, la distance objet-émetteur-récepteur est déterminée à partir de la mesure du temps de propagation de l'onde entre l'émetteur et le récepteur. Il existe plusieurs variantes de la technique de temps de vol en fonction du type de capteurs (calcul par différence de phase). Cette technique est utilisée pour de grandes distances. Contrairement à la triangulation, l'intérêt de cette méthode est qu'elle donne une précision de mesure quasiment constante quelle que soit la distance.

2.4. Les méthodes optiques passives

Cette dernière catégorie regroupe toutes les techniques n'utilisant pas d'éclairage particulier pour faciliter l'extraction de points de la scène. Dans ce cas, il s'agit principalement de la stéréovision passive. La vision stéréoscopique consiste à trouver la profondeur d'un point d'un objet par la mise en correspondance de ce point à l'intérieur de deux images acquises sous des angles différents. La distance du point à la caméra peut être calculée par géométrie épipolaire, à la condition de respecter certaines contraintes sur les points telles les contraintes d'unicité, de continuité et d'ordre.

3. Approches basées sur le diagramme de Voronoï

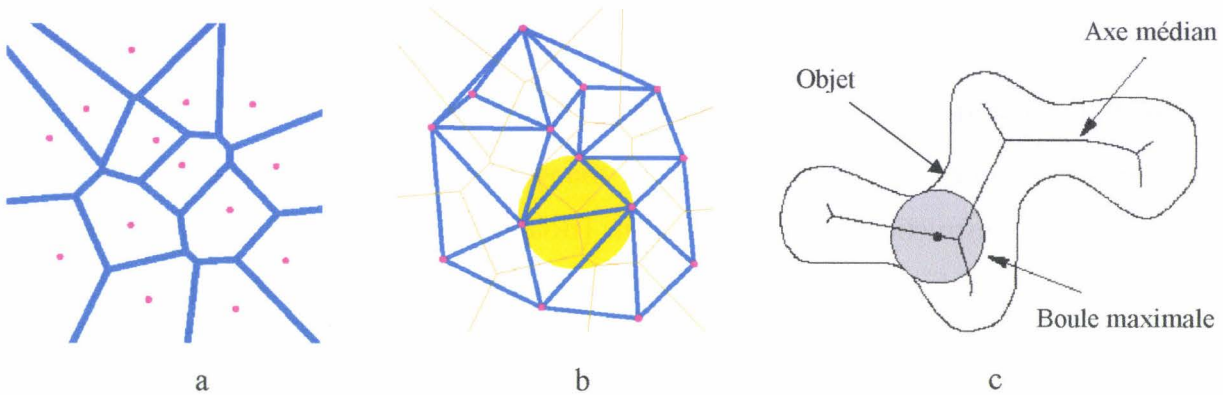
Il existe toute une famille d'algorithmes permettant de générer des surfaces interpolant le nuage de points à reconstruire sous la forme d'un maillage polygonal. Afin de générer le maillage, la plupart de ces approches sont basées sur des algorithmes en deux phases. Une triangulation 3D de Delaunay est tout d'abord réalisée à partir du nuage de points. Ensuite, l'algorithme extrait une surface sous la forme d'un maillage polygonal en sélectionnant les triangles adéquats, sur la base de considérations géométriques et topologiques.

Les maillages polygonaux sont largement répandus dans les applications graphiques, en grande partie parce qu'ils permettent de représenter virtuellement tout type de géométrie et de topologie pour des surfaces de complexité quelconque.

Nous présentons, ci-après, la définition du diagramme de Voronoï et son dual la triangulation de Delaunay, avant d'aborder les principales méthodes utilisant ces approches.

Définition du diagramme de Voronoï :

Soit S un ensemble de N points de $\mathbb{R}^3$. Pour chaque point x de S , la cellule de Voronoï associée à x est définie par l'ensemble des points de l'espace qui sont plus proches de x que de tous les autres points de S . Le diagramme de Voronoï est la décomposition de l'espace formé par les cellules de Voronoï (cf. figure 1.2a). Il permet de représenter la structure topologique d'un nuage de points, et fournit la notion de voisinage d'un point : x est voisin d'un point q de S si et seulement si leur cellules de Voronoï respectives sont adjacentes.



*Figure 1.2 a- Le diagramme de Voronoï sur un nuage de points en 2D,
 b- Triangulation de Delaunay correspondant au diagramme de Voronoï,
 c- Axe médian d'un objet.*

Définition de la triangulation de Delaunay:

La triangulation de Delaunay de S est le dual géométrique du diagramme de Voronoï de S . Deux points de S sont reliés par une arête dans la triangulation de Delaunay si et seulement si leurs cellules sont adjacentes dans le diagramme de Voronoï de S (Figure 1.2b).

La triangulation de Delaunay, aussi appelée tétraédrisation dans l'espace tridimensionnel, est la triangulation qui maximise la compacité des triangles. Il s'agit d'un ensemble de triangles en 2D, ou de tétraèdres en 3D. Elle possède les propriétés suivantes :

- Le maillage de Delaunay interpole l'ensemble des points de S .

- Une arête entre 2 points de S existe si et seulement si leurs cellules de Voronoï sont adjacentes.
- Elle maximise l'angle minimal entre 2 arêtes ayant un sommet en commun.
- Le cercle (sphère) circonscrit dans un triangle (tétraèdre) de Delaunay ne contient aucun autre point de S .

L'axe médian d'un ensemble de points S de $\mathbb{R}^3$ est l'ensemble des points de $\mathbb{R}^3$ n'appartenant pas à S qui ont au moins leurs deux points les plus proches dans S [Blu67] (cf. figure 1.2c). Dans un certain sens, la notion d'axe médian généralise le concept de diagramme de Voronoï d'un ensemble de points pour les surfaces continues [Caz04].

L'axe médian d'un objet (bi ou tridimensionnel) peut être défini comme le lieu des centres des boules maximales incluses à l'intérieur de l'objet, de manière à accéder à une représentation minimale de l'objet considéré. Il peut être démontré qu'une surface lisse peut être reconstruite par l'intermédiaire de l'enveloppe de son axe médian (« Medial Axis Transform »). La notion d'axe médian est à la base de nombreux algorithmes dans le domaine de la reconstruction, du morphing ou de la reconnaissance de formes.

Une des premières références parlant de triangulation de Delaunay utilisable pour des applications de reconstruction de surface est dans [Boi84]. Un très bon recueil des différentes méthodes exploitant le diagramme de Voronoï et son dual pour la reconstruction de surfaces peut être trouvé dans [Caz04].

3.1. Le Crust

Amenta propose la méthode de reconstruction dite du « Crust » [Ame98], « two-pass Voronoï filtering ». Sous certaines conditions d'échantillonnage, l'algorithme du Crust assure la reconstruction correcte de la surface numérisée. Amenta [Ame98] [Ame99] a défini une mesure non uniforme de la densité d'échantillonnage appelée le r -sampling, ou r -échantillonnage, tel que le voisin le plus proche de tout point x de S soit à une distance inférieure à $r \times \text{LFS}(x)$ avec $r < 1$ où $\text{LFS}(x)$, pour « Local Feature Size » d'un point x de la surface S , est la distance euclidienne du point x au point le plus proche de l'axe médian. Intuitivement, une « bonne » densité d'échantillonnage, qui correspond à un échantillonnage suffisamment dense, est inversement proportionnelle à la distance par rapport à l'axe médian.

Dans le cas 3D, le Crust se compose des 4 étapes suivantes :

1. Calcul du diagramme de Voronoï des points échantillons.

2. Calcul des pôles des cellules de Voronoï, définis comme les deux sommets de ces cellules les plus éloignés du point générateur de la cellule. Le premier pôle est le sommet le plus éloigné du point générateur de la cellule, il est noté p^+ , le second pôle noté p^- est le sommet le plus éloigné dans le demi-plan opposé au premier (cf. figure 1.3).

3. On obtient un ensemble de points sommets de cellules de Voronoï, les pôles, qu'on ajoute aux points échantillons de la surface (dans le cas 2D on ajoute tout les sommets des cellules de Voronoï). On calcule la triangulation de Delaunay issue de cette adjonction.

4. On garde uniquement les triangles dont les trois sommets sont des points de l'échantillon de départ. Cette étape est appelée « Voronoï filtering ».

Deux étapes de post traitement ont également été définies (« two-pass Voronoï filtering »), la première étape, « Normal filtering », consiste à supprimer les triangles qui ne remplissent pas le critère de la normale qui spécifie que les normales aux triangles forment de petits angles avec les vecteurs formés par les sommets du triangles et leurs pôles. En utilisant le fait que, sous certaines conditions d'échantillonnage, le vecteur formé par le point x et le pôle p^+ représente une estimation de la normale à la surface au niveau de ce point x (cf. figure 1.3).

La deuxième étape, appelée « Trimming », enlève les petits tétraèdres qui pourraient être laissés sur la surface après le premier post-traitement.

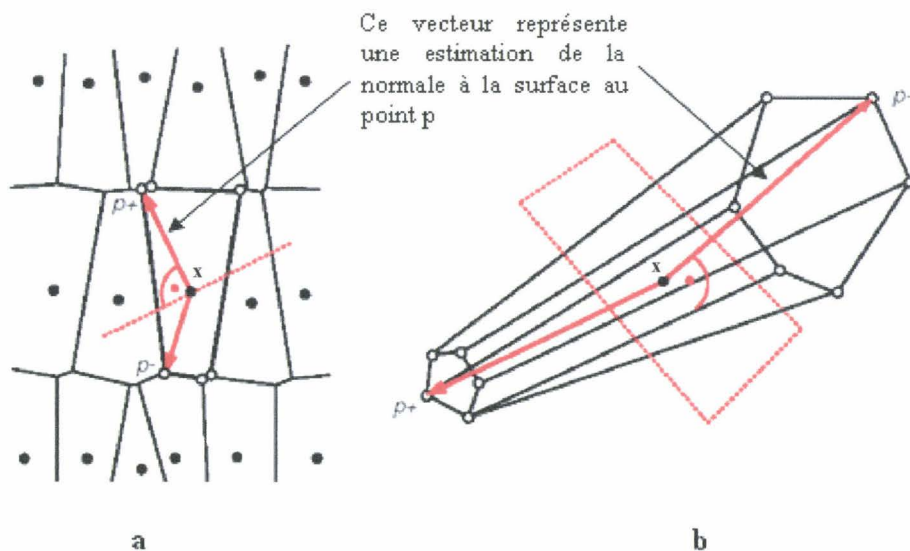


Figure 1.3 Le point x , les sommets des cellules de Voronoï autour de lui et les pôles p^+ et p^- :

a- en 2D,

b- en 3D.

L'algorithme Crust « one-pass filtering » [Ame99] est une approche semblable au Crust « two-pass filtering », il a été proposé par les mêmes auteurs. Il calcule juste le diagramme de Voronoi des points échantillons au lieu des points et des pôles des cellules de Voronoï. Le filtrage est basé sur l'heuristique suivante : l'ensemble de triangles T interpolant la surface S doit satisfaire ces trois conditions :

- a- T contient tous les triangles dont les deux sont les côtés des cellules de Voronoï qui intersectent la surface.
- b- Chaque triangle est petit, c'est-à-dire que le rayon de son cercle circonscrit est beaucoup plus petit que la distance de ses sommets à l'axe médian.
- c- Normal filtering : pour tous les triangles, l'écart angulaire entre les normales aux triangles et les normales estimées en utilisant les pôles aux sommets des triangles doit être faible.

3.2. Le Cocone

Dey [Dey99] propose « Cocone » comme une amélioration du « Crust », basée sur des observations concernant les critères de sélection des triangles faisant partie de la surface finale. Pour chaque point x , on définit 2 cônes alignés sur le plus grand axe de la cellule de Voronoï associée, axe formé par la liaison des deux pôles de la cellule, avec une ouverture de $3\pi/8$, et ayant x pour sommet. Parmi les triangles de Delaunay adjacents au sommet, seuls ceux appartenant au complémentaire de ces 2 cônes sont retenus.

Dey [Dey01] propose d'utiliser Cocone en partitionnant le nuage de points à l'aide d'un octree. Les cellules obtenues sont légèrement agrandies afin d'assurer un recouvrement entre les morceaux de surfaces reconstruites. Grâce à ce partitionnement, Dey arrive à reconstruire explicitement des surfaces de plus d'un million de points sur des machines de puissance modeste en un temps raisonnable.

3.3. Le Powercrust

L'algorithme Powercrust [Ame01], amélioration du Crust, est une technique robuste de reconstruction de surface, qui s'inspire des travaux de Boissonnat [Boi84]. Elle est basée sur une approche duale qui intègre la construction du diagramme de Voronoï et une analyse approximative de l'axe médian.

Le Powercrust construit une approximation linéaire de la surface de l'objet et de son axe médian à partir du nuage de points. L'algorithme utilise le nuage de points pour approcher l'axe médian, puis applique une transformation inverse à partir de cet axe médian pour produire une approximation extérieure de la surface. Les outils principaux de cette méthode sont le diagramme de Voronoï et une sorte de diagramme de Voronoï pondéré appelé « Power diagram ». Néanmoins, cet algorithme est toujours soumis à la condition d'échantillonnage qui stipule que le nuage de points doit être suffisamment dense pour permettre une bonne reconstruction.

Gopi [Gop00] propose de diminuer la dimension du problème en ramenant localement le problème à une reconstruction 2D. Pour cela, les points du nuage doivent être munis d'une normale, afin de déterminer le plan tangent au point. Ensuite, pour chaque point du nuage, on projette son voisinage sur le plan tangent au point. Une triangulation locale de Delaunay permet de générer les triangles incidents au point.

Gao [Gao03] propose de réduire la complexité de la reconstruction en s'appuyant sur le critère de densité minimale (r-échantillonnage) que doit vérifier le nuage de points pour être dans les conditions du Crust [Ame99]. Il propose un re-échantillonnage du nuage de points pour réduire sa taille en ne gardant que les points importants à la préservation de la topologie de la surface à reconstruire.

Le problème des approches basées sur le diagramme de Voronoï leur grande sensibilité au bruit présent dans le nuage de points en entrée, et surtout la complexité calculatoire qui les rend inutilisables pour des nuages de points volumineux, puisqu'elles requièrent la génération complète du diagramme de Voronoï. Par comparaison, notre méthode est basée sur une approche approximative et de ce fait la sensibilité au bruit dans le nuage de points à reconstruire est beaucoup moins critique que dans les méthodes d'interpolation comme les approches basées sur le diagramme de Voronoï. Un autre avantage de notre méthode est qu'elle n'est pas sensible au volume du nuage de points à reconstruire. Notre méthode n'a besoin que des coordonnées 3D des points en entrée, le recours à des informations supplémentaires telles que les normales à la surface ne sont pas nécessaires.

4. Approches implicites

Ces méthodes utilisent une représentation implicite des formes, par opposition à une représentation explicite à l'aide de polygones ou de surfaces paramétriques. Les approches

implicites sont donc définies à partir d'une fonction mathématique $f(x)$ dont on extrait une iso-surface $f(x)=c$ où c est une constante appelée iso-valeur. En général, on s'arrange pour que la surface recherchée soit le lieu où c s'annule. Un parallèle pourrait être fait avec les isobares sur les cartes météo. La fonction $f(x)$ est appelée fonction potentielle, ou champ scalaire. Cette formulation permet a priori de construire n'importe quelle surface à partir d'un polynôme en fonction des coordonnées du point x . Pour illustrer ce principe, un exemple est présenté sur la figure 1.4.

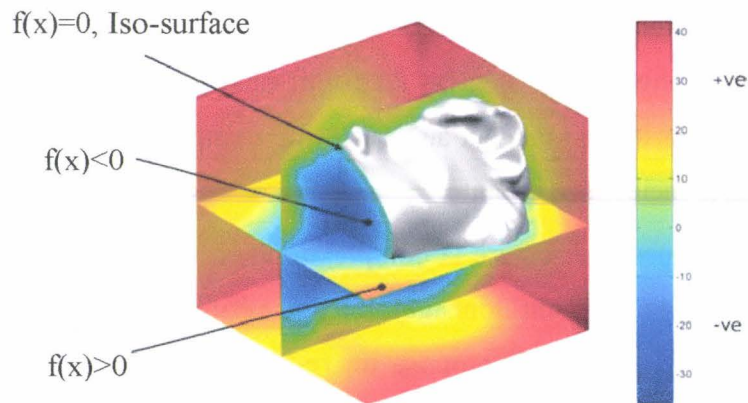


Figure 1.4 Illustration du concept de la représentation d'une surface par surface implicite

Les fonctions implicites peuvent être utilisées pour décrire ou interpoler le comportement d'une forme dans un plan ou dans un espace à trois dimensions.

L'intérêt principal des surfaces implicites est qu'elles permettent de définir tout point x de l'espace en fonction de sa position, il est possible de déterminer si x est à l'intérieur, à l'extérieur ou sur la surface en fonction de la valeur de $f(x)$.

Potentiellement, on dispose d'une infinité de formes et fonctions possibles. Différentes fonctions implicites existent : les surfaces de potentiels [Bli82], les surfaces algébriques [Tau93, Blo97] et les approches variationnelles.

Les surfaces implicites sont une façon élégante de modéliser des surfaces 3D sans avoir à s'occuper des contraintes topologiques. Les surfaces implicites présentent en outre l'intérêt de rendre triviale certaines opérations géométriques parfois coûteuses avec d'autres méthodes de représentation, telles que les opérations booléennes sur des solides (par exemple la différence, l'union ou l'intersection).

Plusieurs approches ont été proposées pour tenter d'interpoler ou d'approcher un nuage de points par une surface implicite, notamment par Hughes [Hug92], Rossignac [Ros94] et He [He94]. On trouvera un tour d'horizon des approches implicites les plus récentes dans [Reu03].

Le problème de la reconstruction d'une surface implicite continue peut être formalisé comme un problème global d'interpolation : à partir d'un ensemble de N points x appartenant à une surface, on cherche à calculer une fonction f telle que $\forall x, f(x)=0$. Comme mentionné par Reuter [REU03], ce problème est sous-contraint puisqu'il existe une infinité de fonctions susceptibles de satisfaire l'équation. Par conséquent, il est nécessaire d'ajouter des contraintes supplémentaires au problème d'interpolation. Ces contraintes vont donc permettre d'obtenir des surfaces implicites d'interpolation, dont différents modèles sont présentés par la suite.

En particulier, les travaux de Turk et O'Brien sur les surfaces implicites d'interpolation [Tur02] constituent une approche relativement robuste pour le problème de la reconstruction de surfaces. Les auteurs proposent ainsi d'utiliser des fonctions à base radiale (RBF) pour satisfaire un ensemble de contraintes définissant la forme. Cette méthode fonctionne à concurrence de quelques centaines de points en entrée, vu qu'elle devient très coûteuse, en temps de calcul, au fur et à mesure que le nombre de points augmente. C'est pourquoi plusieurs auteurs se sont intéressés à la réduction de ces calculs en employant diverses approches.

Nous citerons par exemple les techniques basées sur la notion de partition de l'unité [Fra80] qui permet l'évaluation de la surface implicite issue de plusieurs millions de points. En partitionnant le nuage de points à l'aide d'une structure hiérarchique tel qu'un octree ou arbre BSP (« Binary Space Partition Tree »). Othake [Oth03] proposent d'utiliser la partition de l'unité pour localiser la reconstruction de surfaces quadratiques à l'aide d'un octree, en assumant la connaissance des normales aux points d'entrée.

D'autres méthodes consistent en la reconstruction d'une surface implicite à partir de primitives appelées squelettes [Mur91, Bit95], associées à des fonctions de potentiel, placés de manière cohérente ou aléatoire par rapport aux points en entrée. L'idée principale est de chercher à déterminer une iso-valeur et des primitives tels qu'il existe une iso-surface passant par les points donnés en entrée.

Muraki [Mur91] est le premier à avoir utilisé une surface implicite générée par des squelettes placés au hasard dans le nuage de points de la surface à reconstruire. Chaque point en entrée est muni de son vecteur normal. Des primitives sphériques sont utilisées. A chaque itération, un processus tente de minimiser la différence entre la forme initiale et la surface implicite obtenue. Les résultats sont probants mais cette méthode a une très faible vitesse de

convergence qui la rend inapplicable. Une amélioration notable proposée par Bittar [Bit95] permet de réduire le coût du calcul grâce à l'utilisation de l'axe médian, qui fournit les informations idéales pour choisir plus efficacement où placer les primitives candidates. Un résultat de reconstruction utilisant cette méthode est illustré en figure 1.5.

Malheureusement pour les surfaces implicites, après la reconstruction afin de fournir un rendu de qualité raisonnable, il n'y a généralement pas d'autre choix que de les convertir en maillages polygonaux [Wyv86]. A moins de souhaiter générer des images de synthèse de haut de gamme par lancer de rayon (« Raytracing ») [Har93]. Cette étape est inexistante dans le cas de notre méthode puisqu'elle fournit directement un maillage polygonal approximant la surface, ce qui la rend très facilement utilisable pour des objectifs de rendu de surface.

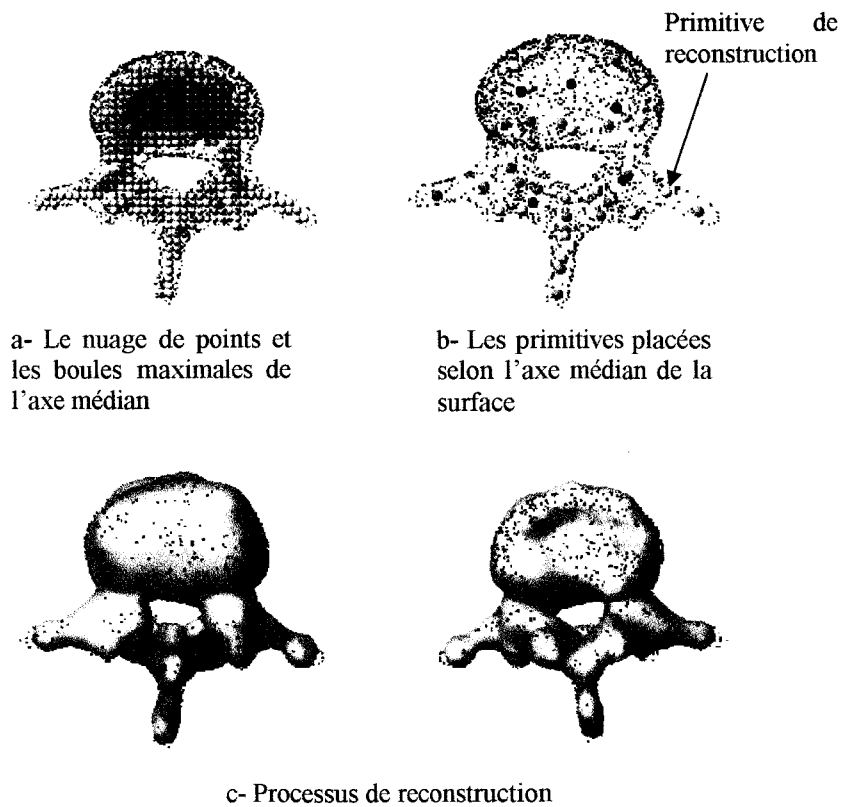


Figure 1.5 Processus de reconstruction d'une surface en utilisant des primitives implicites placées grâce au calcul de l'axe médian

5. Approches mixtes

Dans ce paragraphe nous présentons deux méthodes mixtes. La première méthode est basée sur le calcul d'iso-valeurs, ce qui pourrait lui valoir d'être classée parmi les approches implicites. Néanmoins le calcul d'iso-valeurs n'est qu'un outil de traitement dans le processus de reconstruction de la surface.

La deuxième méthode combine le diagramme de Delaunay et les surfaces implicites comme outils de reconstruction, ce qui permettrait, là aussi, de classer la méthode dans les deux catégories. Dans ce contexte, nous préférons les présenter dans une classe à part, plus globale, que nous appelons « Approches mixtes ».

5.1. Méthode de Hoppe

Il s'agit de l'une des premières méthodes de reconstruction de surface de topologie quelconque, à partir d'un nuage de points non structuré. Comparativement aux autres méthodes, qui supposent la connaissance d'informations supplémentaires telle que l'information d'orientation ou une structuration des points du nuage, Hoppe a proposé dans une série de 3 articles une méthode totalement automatique qui ne requière que les coordonnées tridimensionnelles des points en entrée. Cette méthode n'est pas une interpolation mais une approximation des points en entrée. Cette approche peut se résumer en trois grandes phases :

Phase 1 : Reconstruction d'un maillage triangulaire à partir d'un nuage de points [Hop92]

Le but de cette phase est principalement la détermination de la topologie de la surface. Cette phase de reconstruction d'un maillage primaire à partir d'un nuage de points se décompose en deux étapes :

- Création d'une iso-surface approchant le nuage de points. Pour ce faire, un plan tangent, qui est une approximation locale linéaire de la surface à reconstruire, est estimé pour chaque point à partir de son voisinage. L'iso-valeur calculée en tout point de l'espace est la distance signée au plan tangent du point du nuage le plus proche.
- Extraction des contours du maillage initial à partir de cette iso-surface via une variante de l'algorithme des « marching cubes » [All82] avec l'algorithme de Wyvill et al. [Wyv86]. Le principe de base de l'algorithme « marching cubes » est de subdiviser l'espace en

séries de petits cubes. L'algorithme remplace chacun des cubes avec un ensemble approprié de polygones constituant une approximation linéaire de la surface.

Le maillage triangulaire produit est une surface qui approche celle que le nuage de points en entrée représente. Pour avoir des résultats acceptables, il faut une « très bonne » approximation locale de la normale au plan tangent. C'est l'une des principales limitations de cette méthode.

Phase 2 : Réduction du nombre de triangles du maillage obtenu en première phase [Hop93]

Cette seconde phase permet de réduire le nombre de triangles et optimise la représentation de la surface originale en minimisant une fonction d'énergie basée sur le nombre de sommets dans le maillage et leurs positions dans ce dernier. Il s'agit d'équilibrer le rapport entre une représentation minimale, en terme de nombre de triangles, et une bonne approximation du nuage de points.

Phase 3 : Reconstruction d'une surface de subdivision à partir de ce maillage optimisé [Hop94]

Il est possible de régénérer une surface lisse par morceaux avec le maillage issu de la phase précédente à l'aide d'un algorithme de subdivision minimisant une fonction d'énergie.

Le problème du temps de calcul se pose de manière critique dans cette méthode, car l'augmentation du nombre de points induit une augmentation au moins proportionnelle du temps de calcul, ce qui semble très contraignant dans le cas de nuages de points très volumineux.

5.2. Méthode mixte de Boissonnat

Boissonnat et Cazal [Boi00] proposent la génération d'une surface implicite à partir de l'information de voisinage fournie par le diagramme de Voronoï du nuage de points. Ensuite, ce diagramme est de nouveau utilisé pour produire un maillage adaptatif de la surface implicite. Cette méthode nécessite de disposer de l'information de la normale à la surface pour tous les points, mais elle présente en contrepartie l'avantage d'être moins sensible aux variations de densités de points que les méthodes purement explicites.

6. Approches dynamiques

Certaines approches sont qualifiées de dynamiques du fait qu'en général elles s'appuient sur des maillages qui se déforment au cours du processus de reconstruction en respectant certaines contraintes. Le maillage s'adapte ou évolue tout au long de la procédure pour représenter le nuage de points en entrée.

Moins répandues en 3D, certaines approches dynamiques sont un transfige des algorithmes de type « snakes » ou contours actifs connus dans le domaine de l'analyse d'image. On rappelle ici brièvement le principe des « snakes ».

Définition des snakes :

Les snakes, ou contours actifs, introduits en 1987 par Kass, Witkin et Terzopoulos, [Kas87] sont basés sur les méthodes variationnelles. L'extraction d'une forme à l'aide des modèles déformables (cf. Figure 1.6) est obtenue par la donnée d'une forme initiale et par minimisation d'une fonction d'énergie composée d'un terme de régularisation interne et d'un terme de potentiel d'attraction aux données.

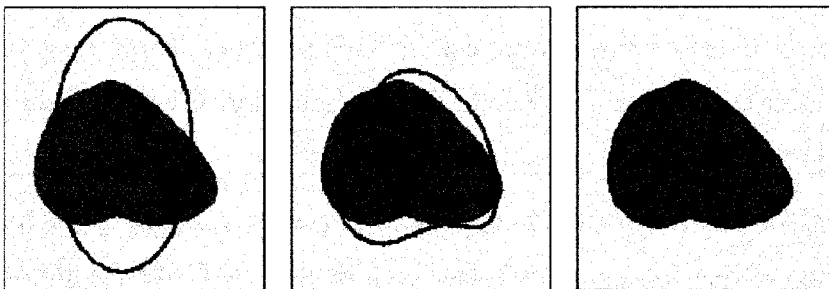


Figure 1.6 Illustration du principe des « snakes » avec les différentes phases de déformation du modèle de « snakes »

En 3D on parlera plutôt de modèle déformable. Celui-ci peut être vu comme une surface qui s'équilibre sous l'influence d'une énergie potentielle. Cette énergie potentielle donne lieu à des forces externes qui déforment le modèle de manière à ce qu'il s'adapte aux données, c'est un objet évolutif. Nous présentons une méthode 3D qui se rapproche du concept des snakes : l'« Intelligent balloon ». Par ailleurs, Plusieurs approches utilisant des modèles ou surfaces déformables ont été développées, par exemple Bardinet [Bar95] s'est servi de modèles paramétriques déformables qui s'adaptent aux points en entrée.

Qin [Qin01] a développé de nombreuses techniques dynamiques de reconstruction de surface. La méthode la plus aboutie est l'« Intelligent Balloon » qui constitue un exemple intéressant de méthode dynamique de reconstruction de surfaces à partir d'un nuage de points.

Le comportement déformable de « l' Intelligent Balloon » est régi par un principe de minimisation d'énergie. Par ce processus de minimisation le modèle « gonflera » à l'intérieur du nuage de points comme un ballon jusqu'à ce qu'il atteigne la frontière de l'objet représenté par ce nuage de points. La structure de départ, appelée aussi graine ou « Seed », (cf. figure 1.7), est représentée par un maillage triangulaire avec peu de facettes (un cube par exemple). La position initiale de la graine est choisie par l'utilisateur à proximité du nuage de points. Qin définit une fonction de coût locale associée à chaque sommet du maillage de la graine. La déformation de ce maillage se fait grâce à la minimisation de cette fonction de coût qui est une combinaison linéaire pondérée de quatre contraintes :

- Le potentiel de déformation représente l'énergie de déformation du modèle. Au cours de chacune des étapes d'évolution, chaque sommet se déplace selon la direction de la normale locale à la structure, afin de faire diminuer son potentiel de déformation.
- La contrainte de bord définit la proximité de chaque sommet par rapport au nuage de points. Cette contrainte de bord est employée pour équilibrer le potentiel de déformation et pour limiter, diriger et contrecarrer la progression générale de la déformation. C'est elle qui permet de décider quand stopper le processus dynamique pour un sommet donné.
- Pendant le processus de déformation, la contrainte de courbure assure qu'aucun sommet ne soit isolé de ses voisins et que les sommets soient bien distribués. Cette contrainte repose sur une approximation de la courbure locale dans le maillage.
- La contrainte angulaire prévient le chevauchement des facettes dans le maillage par la maximisation de l'angle entre deux facettes.

Lorsqu'une facette se stabilise sans diminuer suffisamment son énergie, elle est subdivisée. En quelque sorte, la surface du ballon tend vers la surface définie par le nuage de point.

Avec cette approche, seuls des objets semblables à la primitive graine de départ sont restructurables. Afin de retrouver une topologie arbitraire, un autre test est effectué à chaque étape du processus dynamique : un test de collision inter sommets. A chaque déplacement d'un sommet, on teste son voisinage spatial : si une collision probable est détectée avec un autre

sommet, une connexion est effectuée et la topologie est donc modifiée. Le processus dynamique peut être arrêté lorsque la surface reconstruite satisfait un critère fixé, ou bien à la demande de l'utilisateur.

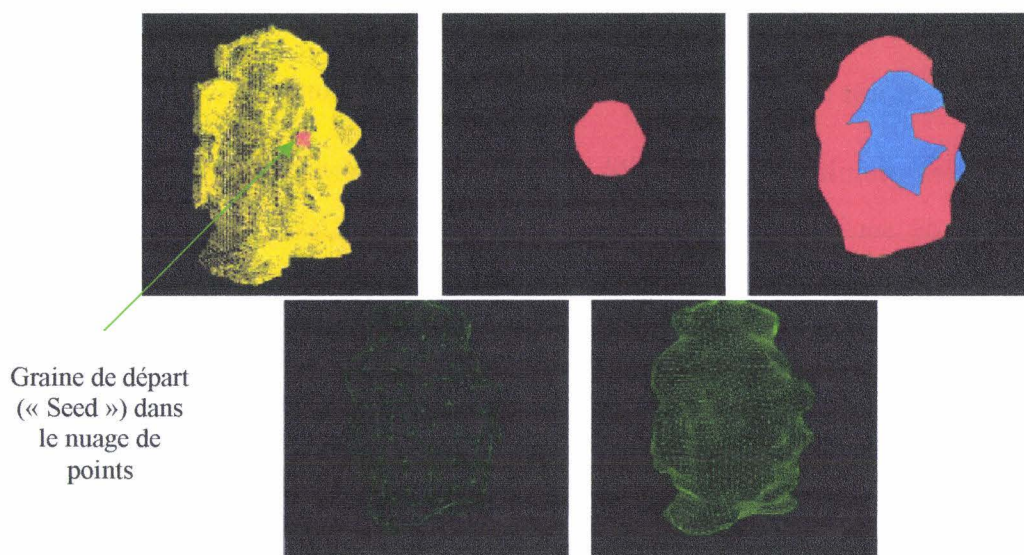


Figure 1.7 Illustration de la méthode de l'« Intelligent Balloon » avec représentation de l'évolution du maillage à chaque étape du processus de la reconstruction

Les résultats obtenus sont assez convaincants, du point de vue complexité topologique. (cf. figure 1.7).

7. Approches neuronales

L'apport des réseaux de neurones aux problèmes de reconstruction de surfaces à partir de nuages de points est relativement récent et s'annonce riche en résultats encore à venir.

Les cartes auto-organisatrices de Kohonen (SOM : « Self Organizing Maps ») [Koh01] représentent l'un des modèles neuronaux les plus utilisés dans le domaine de la reconstruction de surfaces à partir d'un nuage de points. Les SOM sont des modèles neuronaux à apprentissage compétitif, non supervisé. L'une des propriétés intéressante des cartes auto-organisatrices est la faculté de préserver les informations de nature topologiques présentes dans les données.

Nous présentons dans cette partie différentes méthodes de reconstruction de surface à partir de nuages de points utilisant les cartes auto-organisatrices de Kohonen ou d'autres réseaux compétitifs tels que les « Growing Cell Structures » [Fri94].

7.1 Méthodes utilisant les cartes auto-organisatrices de Kohonen

Parmi les premiers à avoir utilisé les cartes auto-organisatrices 2D dans un but de reconstruction de surfaces, on citera Baader et Hirzinger [Baa93], [Baa94]. Leur méthode préconise deux étapes d'apprentissage. Un exemple de reconstruction utilisant une carte auto-organisatrice 2D est illustré en figure 1.8. Cette méthode est limitée à la reconstruction de surfaces 2D.

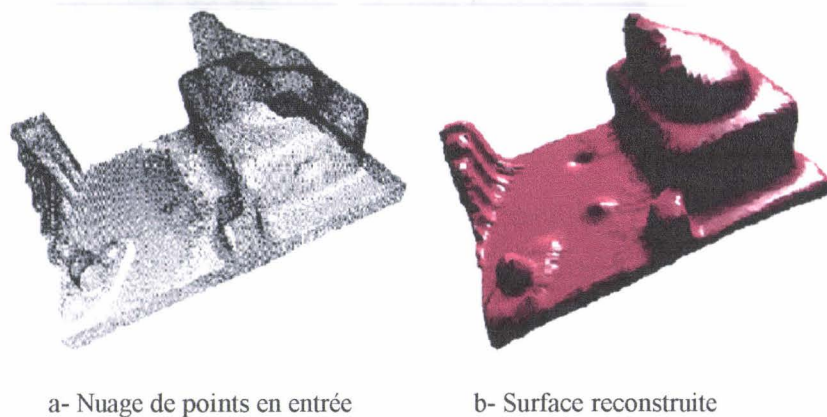


Figure 1.8 Méthode de reconstruction neuronale de Baader et Hirzinger

Barhak [Bar01] a développé une méthode neuronale basée sur les cartes auto-organisatrices 2D à architecture rectangulaire pour créer une grille paramétrique représentant la surface à reconstruire. L'avantage principal de la méthode proposée est qu'elle détecte l'orientation de la grille. La carte auto-organisatrice converge vers la forme du nuage de points en entrée par un apprentissage adaptatif. La méthode est appliquée directement sur les points 3D et évite, selon l'auteur, les anomalies de projection communes aux autres méthodes.

Dans l'approche neuronale de Yu [Yu99] pour arriver à une représentation de l'objet en entrée l'auteur utilise des méthodes de transformation du maillage neuronal. Yu utilise également des méthodes de subdivision générale de l'architecture du réseau pour faire croître la densité neuronale du maillage. Le problème avec cette approche est que certaines méthodes de

transformation du maillage sont exécutées en dehors du processus d'apprentissage ce qui peut être vu comme une supervision partielle du processus.

7.2 Méthodes utilisant les « Growing Cell Structures »

Une autre approche neuronale est celle des « Growing Cell Structures » [Fri94] qui a été mis en place par Ivriissimtzis [Ivr03] à l'institut Max Planck. Une structure de subdivision de l'espace d'entrée est utilisée pour accélérer la phase de recherche du neurone le plus proche du point en entrée, avec un accroissement dynamique de la densité du maillage. Ivriissimtzis préconise une stratégie d'élimination des neurones non actifs via un facteur d'activité qui au fur et à mesure de l'apprentissage accumule les déplacements de chaque neurone et ainsi quantifie son activité. Les neurones avec l'activité la plus faible sont alors supprimés du maillage. L'auteur utilise différentes techniques de transformation de maillage pour améliorer le résultat de la reconstruction [Hop93].

8. Conclusion

Après ce tour d'horizon des diverses approches de reconstruction 3D à partir de nuages de points mettant en relief les avantages et les défauts de chaque approche. Nous allons développer dans le reste de ce mémoire notre méthode qui s'inscrit dans la continuité des méthodes neuronales déjà présentées. Nous proposerons des améliorations des concepts précités en apportant de nouvelles solutions se basant sur des approches originales. Pour initier cette présentation dans le chapitre 2, nous introduisons le concept des cartes auto-organisatrices de Kohonen qui sont à la base de notre démarche.

Les cartes auto-organisatrices de Kohonen

1. Introduction

De nombreuses études ont été menées sur les modèles de représentation cognitifs appliqués aux systèmes perceptifs tels que les systèmes visuels ou auditifs. Ces études ont permis de mettre en évidence des architectures neuronales dédiées et caractérisées par une auto-organisation naturelle des connexions synaptiques suivant une répartition respectant la disposition topologique des régions sensibles du corps humain. Ces études ont, de cette façon, motivé l'introduction des réseaux de neurones formels, appelés « cartes auto-organisatrices ».

Les cartes auto-organisatrices permettent de modéliser quelques fonctionnalités essentielles du cerveau en faisant appel à certains principes fondamentaux du traitement de l'information à l'intérieur du cerveau. Ces cartes ont été développées par T. Kohonen dès 1982 et connaissent aujourd'hui un large essor [Koh 92][Koh 01].

Dans le cadre de l'application des cartes auto-organisatrices de Kohonen au domaine de la reconstruction de surfaces à partir de nuages de points, nous présentons les principales caractéristiques de ces réseaux de neurones, ceci nous permettra de comprendre et de mettre en avant la motivation et les raisons pour lesquelles nous les avons utilisé mais aussi de mettre en exergue les lacunes du modèle actuel. Nous commençons par de brefs rappels biologiques qui nous permettront d'appréhender les principes de base de ces réseaux, puis nous nous intéresserons aux aspects formels de leur représentation.

2. Fondements biologiques

L'étude des aires corticales primaires sensorielles et motrices a permis de déduire des modèles de représentation des organes ordonnés selon un principe de correspondance topologique. L'expérimentation révèle ces aires corticales sous forme de mosaïques fonctionnelles, véritables cartes corticales ordonnées (« ordered feature maps ») qui permettent un décodage efficace des stimuli transcodés par les capteurs périphériques en ce qui concerne les cartes sensorielles, et un adressage précis des ordres de commande vers les effecteurs musculaires pour les cartes motrices. Par exemple, selon les travaux de Hubel et Wiesel [Hub77], dans les aires visuelles, deux colonnes neuronales proches au sein du cortex visuel sont en

correspondance avec deux cellules proches dans la rétine. Des observations identiques ont pu être faites en relation avec le bulbe olfactif, ou l'appareil auditif [Knu79] (voir plus généralement [Woo81]). L'aire de « projection » occupée par chaque région corticale est ainsi représentative d'un type de capteur sensoriel et sa taille est proportionnelle au nombre de capteurs associés.

Dans le traitement de l'information, les cartes corticales réalisent une projection (« mapping ») entre les informations envoyées vers le cerveau sous la forme de signaux biologiques fournis par les capteurs sensoriels périphériques et l'aire corticale associée à la région périphérique sensorielle ou motrice. La rétine se projette ainsi point par point sur le cortex visuel primaire.

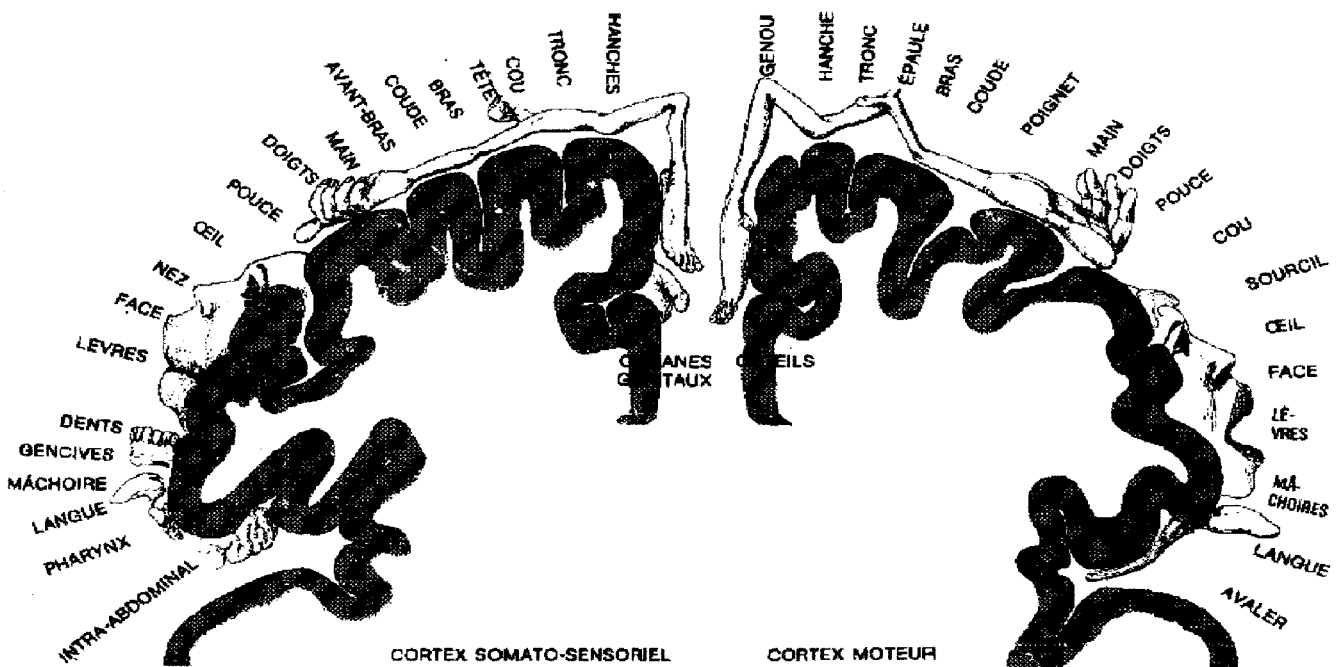


Figure 2.1 Homunculus ou homoncule dessiné par Wilder Penfield et Théodore Rasmussen [Pen50]. La surface occupée au niveau cortical est proportionnelle à la sensibilité sensorielle ou à la précision motrice de la partie du corps correspondante. Ainsi, la surface occupée par le pouce est supérieure à celle correspondant à la cuisse. D'autre part, la topologie est conservée : les doigts sont représentés l'un à côté de l'autre, etc.

Chez l'homme, une cartographie des aires motrices primaires et somato-sensorielles a été établie par Penfield et Rasmussen à l'occasion d'études sur l'épilepsie [Pen50] (cf. figure 2.1). Elle indique très clairement une correspondance topologique entre les parties du corps et leurs projections respectives sur le cortex : deux zones corporelles voisines se projettent sur deux

zones corticales voisines. Chaque partie du corps est représentée en fonction de la surface corticale qu'elle occupe : la surface corticale est proportionnelle à l'importance fonctionnelle du segment corporel représenté. De telles propriétés sont également vérifiées pour le cortex visuel et on dit, dans ce cas, que des neurones voisins à l'intérieur de la rétine correspondent à des connections synaptiques entre neurones voisins à l'intérieur du cortex cérébral (neurones tectaux). On parle d'organisation rétinotopique ou de carte rétinotopique.

Von der Malsburg est l'un des premiers à avoir mené des travaux sur la modélisation de l'auto-organisation à partir d'observations faites sur l'organisation corticale. Il cherchait à comprendre la structuration du module fonctionnel formé par des colonnes d'orientation du cortex visuel composées de neurones réceptifs à la même orientation du stimulus [Mal73]. Par la suite, il étudia les mécanismes des cartes rétinotopiques avec Willshaw [Wil76]. Plus tard, Takeuchi et Amari ont proposé une analyse mathématique d'une version révisée du modèle rétinotopique de Willshaw-Malsburg [Ama79].

De façon générale, toutes les voies sensorielles majeures préservent l'organisation spatiale des récepteurs correspondants à l'intérieur du cortex cérébral. Il s'agit d'une propriété majeure des cartes auto-organisatrices qui permet de garder une correspondance de voisinage entre l'espace d'entrée des données et l'espace de représentation : deux données « proches » de l'espace d'entrée sont projetées à l'intérieur d'une même région dans l'espace de représentation.

L'expérimentation montre qu'il est très improbable que les connexions qui participent à ces projections topiques soient entièrement déterminées génétiquement, même si elles le sont en grande partie. En effet, le système nerveux se structure au cours du développement de l'individu et, jusqu'à une époque récente, sa structure était considérée comme relativement stable chez l'adulte. Des études récentes ont cependant montré que le cerveau manifeste une remarquable capacité à s'adapter aux évolutions importantes de son environnement en élaborant de nouveaux chemins privilégiés afin d'assurer un meilleur transfert des informations importantes. Cette plasticité cérébrale se manifeste notamment au niveau du cortex sensoriel [Xer98] et se concrétise par une adaptation continue des connexions synaptiques sollicitées. Par exemple, plus un violoniste s'entraîne, plus la zone du cortex liée à son auriculaire gauche s'étend au détriment des zones allouées aux autres doigts, ce qui rend ce petit doigt d'autant plus agile ! Cette adaptation des connexions inter-neurones est particulièrement active chez les jeunes individus car elle correspond spécifiquement à un processus d'apprentissage.

L'étude des phénomènes biologiques au niveau du cerveau a apporté un grand nombre d'informations sur son fonctionnement, sur celui de l'élément fonctionnel de base du système

nerveux qu'est le neurone biologique et sur la manière dont fonctionnent les réseaux de neurones biologiques [Bru90]. Dès lors, des mathématiciens ont tenté de reproduire le fonctionnement du cerveau en intégrant ces connaissances de nature biologique dans des programmes informatiques, et en leur donnant la capacité d'acquérir de nouvelles connaissances. Cela a commencé en 1943 avec Mc Culloch et Pitts [Mcc43]. Leurs travaux ont débouché sur ce qu'on appelle aujourd'hui communément les "réseaux de neurones artificiels". Il s'agit de comprendre les principes selon lesquels les systèmes biologiques traitent les signaux et de s'en inspirer pour élaborer de nouvelles techniques de traitement de l'information applicables dans le champ des sciences de l'ingénieur [Ros58] [Min69] [Hop82] [Rum86] [Koh81a, b, c, d]. Certes, si copier le cerveau restera pour longtemps encore un objectif très ambitieux et difficilement atteignable, vouloir s'inspirer des architectures et des fonctions du système nerveux n'est cependant pas un rêve inaccessible.

La brique élémentaire des réseaux de neurones artificiels est le neurone formel qui constitue une modélisation fonctionnelle et mathématique des fonctions essentielles du neurone biologique [Mcc43] (cf. figure 2.2). Chaque neurone formel, dit neurone artificiel, peut être considéré à ce titre comme un processeur élémentaire. Il dispose d'un nombre variable d'entrées. On associe à chacune d'entre elles un poids de connexion représentatif de la plasticité de la liaison entre l'entrée et le neurone. L'ensemble des poids en entrée du neurone est appelé vecteur-poids du neurone. La sommation des entrées pondérées par leurs poids respectifs représente le potentiel d'activation du neurone. C'est lui qui précise le niveau d'excitation du neurone face à une entrée donnée et qui va induire une réponse sur la sortie du neurone après avoir été pris en compte par la fonction de transfert du neurone. Il existe de nombreuses formes de fonctions de transfert, aussi appelées fonctions neurone ou fonctions d'activation, telles les fonctions identité, signe, seuil, linéaire, sigmoïde...

Chaque neurone d'indice i est identifié par son vecteur-poids m_i , l'entrée x et sa sortie s_i . En considérant que chaque neurone possède n entrées distinctes, nous aurons $x = \{x_1, x_2, \dots, x_n\}$ et $m_i = \{m_{i1}, m_{i2}, \dots, m_{in}\}$. La sortie s_i permet de délivrer le résultat du calcul réalisé par la fonction de transfert, notée f , telle que $s_i = f(x, m_i, \mu)$, où μ représente un ensemble de paramètres fixés par l'utilisateur.

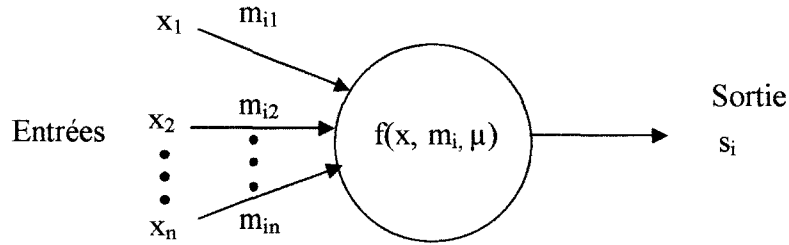


Figure 2.2 Modèle du neurone formel

L'association de plusieurs neurones artificiels reliés entre eux grâce à leurs connections en entrée et en sortie permet d'aboutir à la structure essentielle que représente le réseau de neurones artificiels.

L'intérêt d'organiser les neurones en réseau réside principalement dans le fait que, de cette combinaison, naissent des propriétés remarquables propres à ces réseaux, telles que la capacité d'apprentissage ou encore celle d'adaptation. Les connexions entre les neurones qui composent le réseau décrivent la « topologie du réseau ». Elle peut être quelconque mais, le plus souvent, les neurones sont arrangés par couches. L'interconnexion entre les neurones des différentes couches permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et ainsi de définir les concepts de « neurone d'entrée » et de « neurone de sortie ». Par extension, on appelle couche d'entrée la couche formée par l'ensemble des neurones d'entrée et couche de sortie la couche formée par l'ensemble des neurones de sortie.

L'apprentissage est un processus dynamique et itératif qui se caractérise par une modification de l'efficacité synaptique, c'est-à-dire par un changement des valeurs des poids des connexions synaptiques en fonction des données qui sont présentées successivement en entrée du réseau. Un ensemble de règles précises permettant de réaliser un tel processus d'apprentissage par adaptation des poids des neurones représente l'algorithme d'apprentissage du réseau.

La base d'apprentissage d'un réseau de neurones est un ensemble de données, représentées dans un espace d'entrée.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes familles selon que l'apprentissage est supervisé ou non-supervisé. Cette distinction repose sur la forme de la base d'apprentissage. Dans le cas de l'apprentissage supervisé, les données présentées au réseau sont des couples (Entrée) / (Sortie désirée), tandis que dans le cas d'un apprentissage non-supervisé on ne dispose que des valeurs (Entrée) sans connaître les sorties qui doivent leur correspondre.

Le type d'apprentissage est déterminé par la manière dont les poids synaptiques sont modifiés. L'apprentissage compétitif est un processus qui a été mis en évidence dans le cadre des réseaux biologiques [Hay94]. Il s'avère que les réseaux de neurones artificiels qui se basent sur un apprentissage non-supervisé compétitif sont les modèles qui se rapprochent le plus des réseaux neuronaux biologiques présents dans le cerveau. Ainsi les réseaux compétitifs de Kohonen semblent correspondre fortement aux architectures neuronales présentes dans le cortex visuel et le processus d'adaptation du réseau par apprentissage compétitif semble modéliser étroitement les phénomènes d'adaptation qui se produisent dans le cortex visuel peu de temps après la naissance.

3. Réseaux compétitifs

Les réseaux compétitifs sont basés sur un mode d'apprentissage qui stimule la compétition entre les neurones dans le but de favoriser l'adaptation des neurones les plus réactifs aux données présentées en entrée du réseau.

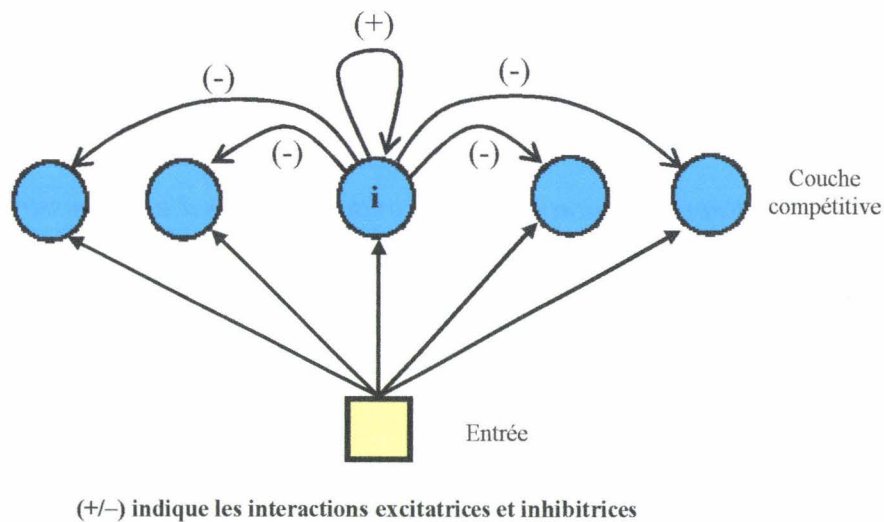


Figure 2.3 Architecture d'un réseau compétitif

Dans leur forme la plus simple, les réseaux compétitifs sont souvent constitués de deux couches. Une couche de neurones sur laquelle sont présentées en entrée les données issues de la base d'apprentissage, il s'agit de la couche d'entrée du réseau. Une couche de sortie qui permet de restituer globalement les résultats des calculs fournis par le processus compétitif (cf. figure 2.3). Cette couche de sortie est également appelée couche compétitive. Le processus compétitif entre les neurones est réalisé traditionnellement grâce à des connexions latérales inhibitrices. Ces

connections latérales entre neurones voisins permettent à chaque neurone de sortie d'augmenter naturellement son niveau d'activation tout en inhibant le niveau d'activation de ses voisins. Une telle architecture, couplée à un apprentissage compétitif, permet de respecter, au cours de sa phase d'adaptation, l'organisation topographique des données dans l'espace d'entrée.

3.1. Stratégie d'apprentissage compétitif (« Winner Takes All »)

Contrairement aux autres types d'apprentissage où, généralement, tous les neurones peuvent apprendre simultanément et de la même manière, l'apprentissage compétitif désigne un « vainqueur », et seul celui-ci bénéficie d'une adaptation. Une telle stratégie est appelée « tout au vainqueur ». Elle est notée : WTA pour « Winner Takes All ».

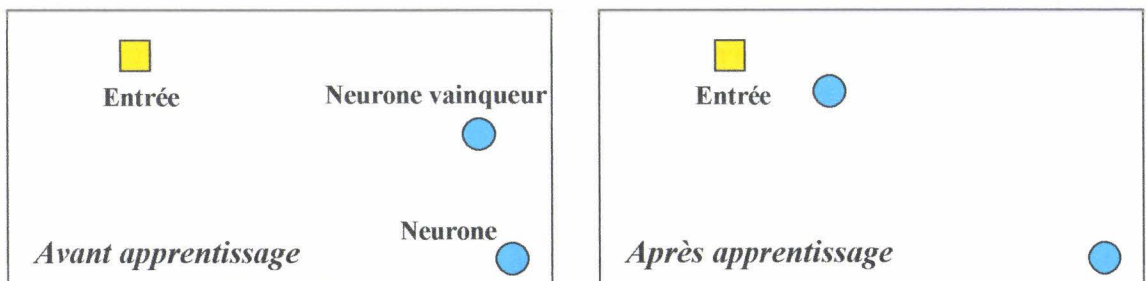


Figure 2.4 Représentation dans un plan du processus d'apprentissage pour le neurone vainqueur par rapport à une donnée présentée en entrée. La position du neurone vainqueur dans l'espace d'entrée est directement donnée par son vecteur-poids. Après adaptation, le nouveau vecteur-poids révèle que le neurone gagnant s'est rapproché de la position de la donnée présentée en entrée.

Le processus d'apprentissage se déroule en trois étapes :

- Sélection et présentation d'une donnée sur la couche d'entrée.
- Les neurones de la couche compétitive réagissent à l'entrée. Le neurone ayant la plus grande activité est déclaré vainqueur et seul ce neurone reste actif. Tous les autres neurones sont inhibés [Rum85], [Lip87].
- Après sélection du neurone vainqueur, seul celui-ci bénéficie de l'adaptation.

Les vecteurs-poids des neurones et des vecteurs d'entrée sont de même dimension. De ce fait, les neurones du réseau peuvent être représentés dans l'espace d'entrée.

Au cours de l'apprentissage compétitif, l'adaptation du neurone vainqueur se fait en modifiant son vecteur-poids de façon à ce qu'il se rapproche de la valeur du vecteur d'entrée à qui il doit sa victoire (cf. figure 2.4). De ce fait, les chances de victoire de ce neurone seront plus élevées quand on présentera la même donnée, ou une autre donnée proche de la première. Par contre, il sera relativement défavorisé quand d'autres données, plus éloignées, seront présentées au réseau.

Ce type d'apprentissage est utilisé comme moyen de catégorisation des données présentées en entrée du réseau. En effet, quand on présente celles-ci un nombre suffisant de fois, chaque neurone du réseau acquiert une sensibilité différente par rapport aux données [Gro76], [Gro88], [Koh84]. Au cours de l'apprentissage, chaque neurone devient représentatif d'un sous-ensemble restreint d'entrées dont les similitudes peuvent être globalement représentées par un neurone unique dont le vecteur-poids constitue le vecteur prototype de ce sous-ensemble.

3.2. Stratégie d'apprentissage compétitif (« Winner Takes Most »)

Dans le principe d'apprentissage compétitif « Winner Takes All », seul le gagnant est adapté à chaque itération. Les algorithmes basés sur ce principe d'apprentissage compétitif présentent des vitesses de convergence relativement faibles, surtout lorsque le nombre de neurones est grand, car chaque entrée présentée ne déclenche l'adaptation que d'un seul neurone. Il apparaît donc souhaitable de disposer d'algorithmes où plusieurs neurones sont adaptés simultanément à chaque itération. Ce type d'adaptation impliquant plusieurs neurones à la fois repose en général sur la notion de voisinage topologique entre neurones à l'intérieur de l'architecture du réseau. Le problème est de définir avec soin la fonction de voisinage qui intervient pour le neurone vainqueur dans la règle d'adaptation, c'est-à-dire de définir quels sont les neurones voisins du neurone gagnant qui sont adaptés en même temps que ce dernier. Ce type d'apprentissage compétitif est appelé « Winner Takes Most » ou WTM.

Von der Malsburg [Mal73], dans le cadre de ses travaux sur la modélisation des fonctions cérébrales, utilise un réseau complètement interconnecté composé de deux couches. La couche d'entrée est composée de capteurs visuels, organisés sous la forme d'une grille bidimensionnelle de P pixels. La couche de sortie prend également la forme d'une grille bidimensionnelle où chaque neurone possède un vecteur-poids de dimension n . Chaque pixel de la couche d'entrée est connecté à tous les neurones de la couche de sortie. Au sein du réseau, des connexions latérales inter-neurones sont excitatrices pour les courtes distances et inhibitrices pour les longues

distances. Le poids de ces connexions est établi en fonction de la distance et suivant un profil en chapeau mexicain (« Mexican hat ») (cf. figure 2.5). Ainsi, dans ce modèle, le neurone vainqueur n'est plus le seul à subir le processus d'apprentissage, ses voisins y sont également soumis.

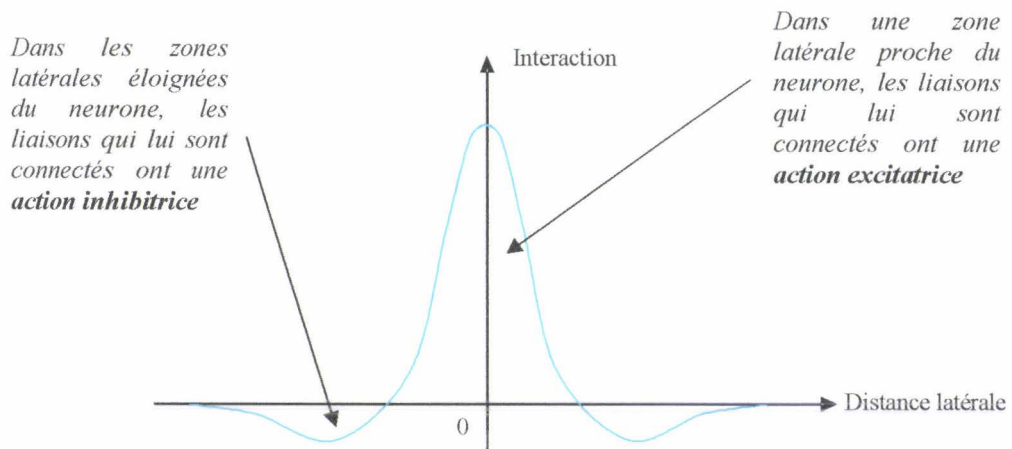


Figure 2.5 Fonction d'interaction latérale dite du « chapeau mexicain »

Lorsqu'une entrée est présentée au niveau de la couche d'entrée du réseau, les neurones montrent une réactivité plus ou moins forte en fonction de l'adéquation entre l'entrée elle-même et leurs poids. Le neurone gagnant est celui qui révèle le maximum d'activation. Un processus de coopération / compétition se met alors en place au moyen des connexions latérales inter-neurones placées entre le neurone gagnant et ses voisins.

Le but de cet apprentissage compétitif de type WTM est de permettre au réseau d'adapter les poids des connections de telle sorte que des données présentées en entrée du réseau issues de plusieurs capteurs voisins de la rétine puissent être projetées en une zone du réseau impliquant plusieurs neurones voisins au sens de la topologie du réseau. Cette auto-organisation topologique du réseau permet de rendre compte de l'organisation et de la distribution spatiale des données dans l'espace d'entrée.

3.3. Extensions et développement des réseaux compétitifs

Nous présentons ici quelques réseaux compétitifs parmi les plus utilisés. Cette présentation nous permettra d'appréhender ultérieurement de façon synthétique l'étude des

réseaux compétitifs de Kohonen dont nous nous inspirons fortement dans le cadre de notre travail de recherche.

Fukushima a été l'un des premiers à s'inspirer de l'organisation biologique des réseaux et de leur processus d'apprentissage dans le contexte d'une application [Fuk75]. Il a initialement développé le *Cognitron*, un réseau multicouche à organisation automatique. Ce réseau était d'une grande complexité et était destiné à la reconnaissance d'écriture manuscrite dans le but de mieux comprendre les mécanismes effectivement mis en œuvre pour réaliser efficacement ce processus de reconnaissance.

Développé en 1987 par Carpenter et Grossberg [Car87a], le réseau ART (« Adaptive Resonance Theory ») est un modèle de réseau de neurones compétitifs à architecture évolutive permettant de créer un système d'apprentissage autonome capable de concilier adaptation et stabilité grâce à un compromis entre plasticité et rigidité synaptique. Il existe plusieurs versions pour ce réseau : ART, ART2 [Car87b], ART3 [Car90]. Dans ce type de réseau, les vecteurs-poids ne sont adaptés que si l'entrée est suffisamment proche d'un prototype déjà connu par le réseau : on parle alors de résonance. A l'inverse, si l'entrée est trop éloignée des prototypes existants, une nouvelle catégorie est créée, avec pour prototype l'entrée qui a engendré sa création.

Les cartes auto-organisatrices représentent un autre exemple de réseaux de neurones compétitifs à apprentissage non supervisé. Elles ont été introduites par Tuevo Kohonen au début des années 80 [Koh81 a, b, c, d], [Koh82 a, b]. Leurs principales caractéristiques sont de générer une interaction compétitive entre les neurones grâce à la notion de voisinage d'adaptation. Une compression de l'information est réalisée grâce à une projection non-linéaire des données de l'espace d'entrée dans l'espace de sortie. Cette projection possède la propriété de préserver la topologie de l'espace d'entrée : des données proches dans l'espace d'entrée ont des représentations proches dans l'espace de sortie.

Les cartes auto-organisatrices se présentent en général sous la forme d'une architecture neuronale organisée suivant une configuration 2D, telle une grille bidimensionnelle où les mailles représentent des liens de voisinage topologique entre les neurones et où chaque sommet de maille représente un neurone particulier.

Les cartes de Kohonen sont souvent utilisées dans le cadre de la projection de données multidimensionnelles, de la classification de données, ou encore de l'optimisation de solutions de problèmes, tel celui du voyageur de commerce. Elles ont été utilisées avec efficacité en reconnaissance de la parole [Koh al. 84], traitement d'images [Her94], robotique [Rit92],

contrôle de processus [Try91], visualisation de données, « Data minning » [Kas95], « Web minning » [Hon97], analyse médicale [Che03] ou encore en analyse financière [Ekl03]. Une palette exhaustive des applications des cartes auto-organisatrices est disponible dans l'ouvrage de référence écrit par Kohonen [Koh02].

Plus récemment sont apparus les réseaux dits « Neural Gas », développés par Martinetz et Schulten [Mar91]. Dans ce type de réseau, les neurones sont ordonnés en fonction de la distance entre le vecteur-poids du neurone vainqueur et la donnée présentée en entrée. Les neurones ne sont pas interconnectés par des liens topologiques. Seuls les neurones les plus proches de la donnée sont adaptés. Le nombre de neurones modifiés ainsi que l'ampleur de cette modification diminuent avec le temps.

Le réseau « Growing Neural Gas » de Fritzke [Fri95] est une variante du *Neural Gas* où, cette fois-ci, les neurones sont interconnectés. Un réseau évolutif minimal est initialement créé et de nouveaux neurones ainsi que de nouvelles connexions entre les neurones sont ajoutés au cours de l'apprentissage qui reste non supervisé.

4. Carte auto-organisatrice de Kohonen

Dans ce paragraphe, nous abordons une présentation plus approfondie des réseaux de Kohonen car ils représentent, tant au niveau de l'architecture qu'au niveau du processus d'apprentissage, la base de notre approche.

L'étude des fonctionnalités biologiques du corps humain révèle en particulier que le système nerveux semble en partie se développer en l'absence de plan et d'instructions fixés au préalable. Ces extraordinaires fonctionnalités sont en partie dues à la capacité d'auto-organisation des liaisons nerveuses via une adaptation de la plasticité synaptique, ce qui induit une remarquable efficacité pour le traitement des informations.

Dans ce contexte, les simulations faites par de nombreux auteurs ont démontré que, pour obtenir un comportement d'auto-organisation efficace et semblable à celui que l'on peut trouver dans le cortex cérébral, il faut respecter les deux principes suivants :

- Les neurones de la couche compétitive doivent réagir différemment aux entrées afin de pouvoir élire un neurone vainqueur.
- Durant le processus compétitif, l'adaptation synaptique des liaisons doit être distribuée à l'intérieur d'un voisinage centré sur le neurone vainqueur.

Kohonen [Koh82b], [Koh84] proposa une généralisation et des simplifications algorithmiques du modèle neuronal de Von der Malsburg [Mal73]. Le mécanisme relativement complexe de formation des zones d'activation et d'inhibition par le jeu de connexions latérales dont les poids sont fixés sur la base de valeurs suivant le profil d'une fonction de type « chapeau mexicain » est remplacé par la simple prise en compte d'une zone d'adaptation. Il n'y a plus de connexions latérales « fixes » entre neurones et on procède à une simple détermination du neurone vainqueur. L'adaptation se fait alors pour tous les neurones placés autour du neurone gagnant et à l'intérieur d'un voisinage d'adaptation défini au sens de la topologie du réseau et non en terme de distance entre les poids des neurones représentatifs de la position de chaque neurone dans l'espace de représentation des données en sortie du réseau.

Un autre apport de l'approche proposée par Kohonen par rapport au modèle de Von der Malsburg est de remplacer, pour détecter le neurone vainqueur, la recherche du maximum d'activité basée en pratique sur le produit scalaire entre la donnée x présentée en entrée et le vecteur-poids m_i du neurone d'indice i , par la recherche de la distance minimale entre la donnée x présentée en entrée et les vecteurs-poids m_i des neurones du réseau. Dans le cas d'une distance euclidienne avec des vecteurs-poids m_i normés :

$$\|x - m_i\|^2 = \|x\|^2 - 2x^T m_i + \|m_i\|^2$$

En effet, puisque la norme de l'entrée x est indépendante du neurone i auquel elle est présentée et puisque la norme du vecteur-poids m_i est constante quand les poids sont normalisés, le neurone pour lequel le produit scalaire $2x^T m_i$ est maximum correspond au neurone le plus proche au sens de la distance euclidienne. Le carré de la norme convient également, sachant qu'il faut trouver le minimum, ce qui évite le calcul de la racine carrée. L'avantage d'utiliser une distance au lieu du produit scalaire réside principalement dans le fait que cette approche ne nécessite pas de phase préalable de normalisation des vecteurs-poids.

4.1. Architecture et topologie de la carte de Kohonen

L'architecture neuronale d'une carte de Kohonen est essentiellement composée d'une couche d'entrée et d'une couche compétitive (cf. figure 2.6). L'organisation topologique de la carte est établie à partir de la relation de voisinage définie entre les neurones du réseau.

- La couche d'entrée sert uniquement à la présentation des données x au réseau.

- La couche compétitive est très souvent formée de neurones disposés suivant une organisation topologique en forme de grille 2D mais il existe également des formes topologiques 1D et 3D. Dans tous les cas, la position de chaque neurone est repérée par rapport aux autres neurones à l'intérieur de l'architecture du réseau. Dans le but de favoriser le processus de compétition et d'auto-organisation entre les neurones durant la phase d'apprentissage, les données présentées en entrée du réseau sont systématiquement recopiées sur l'entrée de chaque neurone qui délivre, après calcul, une valeur d'activation sur sa sortie. Les fonctions d'activation généralement utilisées sont des fonctions linéaires.

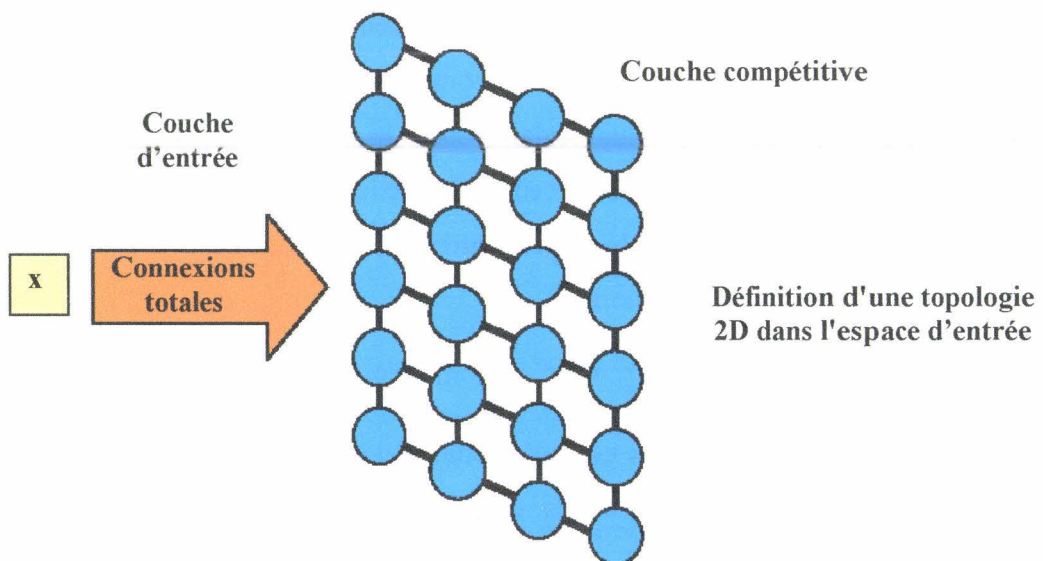


Figure 2.6 Architecture neuronale de la carte auto-organisatrice 2D

La figure 2.7 illustre quelques exemples d'architectures. Elles peuvent être 1D, 2D rectangulaire ou hexagonale, voire même 3D cubique.

L'organisation topologique des neurones à l'intérieur du réseau suivant un ordre établi nécessite de définir la notion de distance topologique entre les neurones. Le calcul de cette distance est spécifique à chaque organisation topologique du réseau. Il permet de spécifier les différents niveaux de voisinage pouvant exister entre un neurone et ses voisins.

Par exemple, dans le cas d'une carte 2D rectangulaire, chaque neurone est associé à un vecteur à deux dimensions dont les deux composantes indiquent sa position dans la carte 2D. Pour définir le niveau de voisinage auquel appartient un neurone quelconque d'indice j par rapport à un neurone particulier d'indice i , on calcule une distance topologique inter-neurones qui est définie pour une carte 2D par : $d_v(i,j) = \max(|r_{i1} - r_{j1}|, |r_{i2} - r_{j2}|)$; où $r_i(r_{i1}, r_{i2}) \in \mathbb{R}^2$ et $r_j(r_{j1}, r_{j2}) \in \mathbb{R}^2$

indiquent les positions des neurones d'indices i et j dans la carte. Pour illustrer cette notion de distance topologique, nous pouvons considérer l'exemple de la figure 2.8 où la distance entre les neurones d'indices i et j est $d_v(i,j)=2$. Dans ce même exemple tous les neurones se trouvant à une distance topologique de valeur 2 par rapport au neurone d'indice i appartiennent à ce qu'on appelle son deuxième niveau de voisinage. Il en va de même pour les différents niveaux de voisinage, chaque niveau est défini par la distance entre chaque neurone et ses voisins immédiats ou lointains.

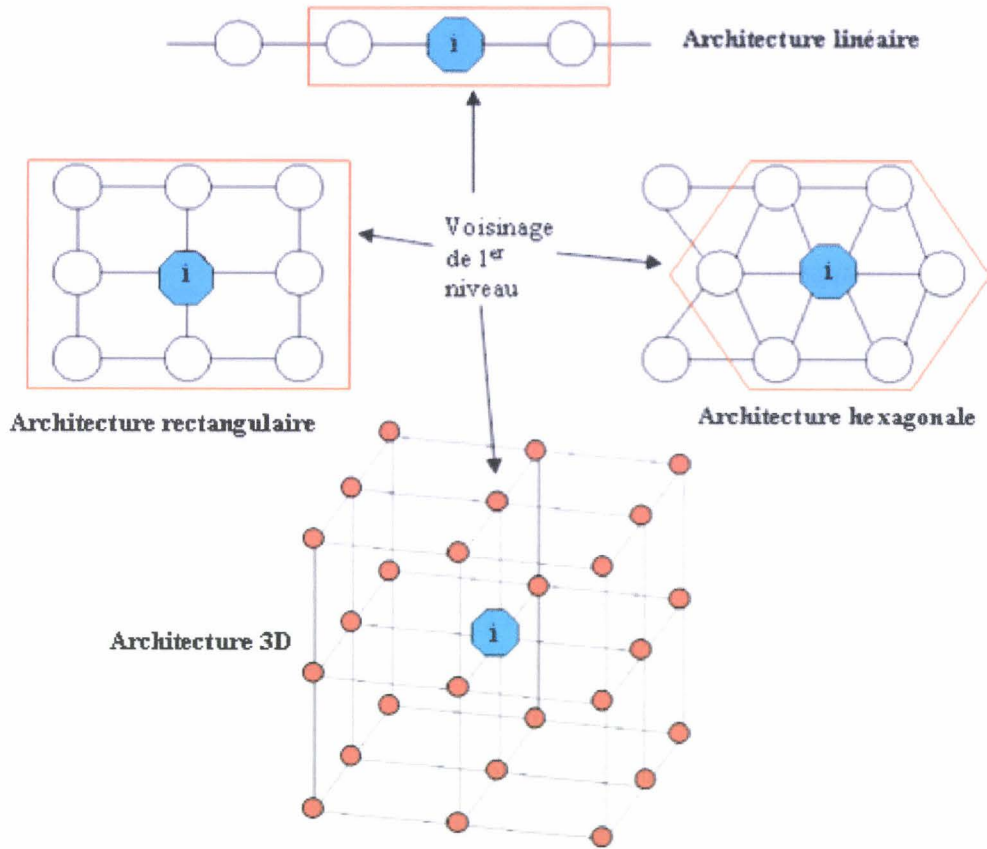


Figure 2.7 Illustration de la notion de voisinage de 1^{er} niveau pour différents types d'architectures

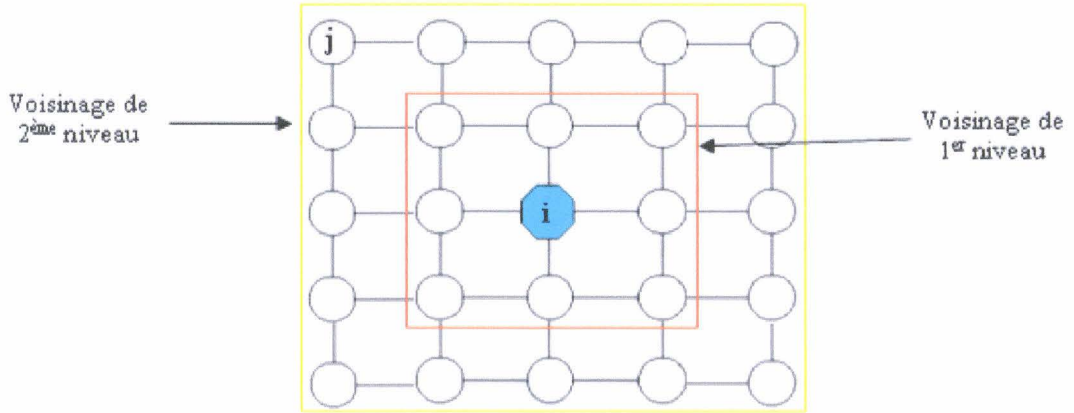


Figure 2.8 Illustration de la notion de voisinage topologique

4.2. Principe général de l'algorithme de Kohonen

4.2.1. Aspects compétitifs

A chaque neurone d'indice i de la carte est associé un vecteur-poids $m_i = [m_{i1}, m_{i2}, \dots, m_{in}]^T \in \mathcal{R}^n$, $i=1,2,\dots,M$, M étant le nombre total de neurones dans le réseau. A chaque itération de la procédure d'apprentissage, un vecteur d'entrée $x_j = [x_{j1}, x_{j2}, \dots, x_{jn}]^T \in \mathcal{R}^n$, avec $j=1,2,\dots,N$, est choisi aléatoirement parmi les données appartenant à l'ensemble d'apprentissage, puis est présenté à l'entrée de tous les neurones. N représente le nombre de données dans la base d'apprentissage. L'entrée x_j choisie à l'itération de rang t est noté $x(t)$. t est un entier qui représente le rang de l'itération du processus d'apprentissage.

La compétition entre les neurones oblige, dans un premier temps, à comparer les distances entre le vecteur d'entrée $x(t)$ et tous les vecteurs-poids $m_i(t)$ afin de repérer le neurone le plus proche de la donnée présentée en entrée. La norme euclidienne $d_e(.)$ est généralement retenue comme mesure de distance entre le vecteur-poids du neurone d'indice i et le vecteur d'entrée $x(t)$:

$$d_e(x(t), m_i(t)) = \|x(t) - m_i(t)\|.$$

Le neurone qui minimise cette distance est déclaré neurone gagnant. Il est alors repéré par l'indice « c » et son vecteur-poids est noté m_c :

$$\|x(t) - m_c\| = \min_i \{\|x(t) - m_i(t)\|\}.$$

4.2.2. Aspects liés à l'apprentissage

Durant l'apprentissage, le neurone gagnant, ainsi que ses voisins au sein de la carte, vont s'adapter en partie au vecteur d'entrée $x(t)$. Il en résulte pour les vecteurs-poids des neurones appartenant au voisinage d'adaptation du neurone gagnant un phénomène de rapprochement puisqu'ils s'adaptent par rapport au même vecteur d'entrée $x(t)$. Cette adaptation va au fur et à mesure de la présentation des vecteurs d'entrée x , se traduire par une organisation globale des neurones vis-à-vis de l'ensemble d'apprentissage. La loi d'apprentissage appliquée aux vecteurs-poids du neurone gagnant et de son voisinage est :

$$m_i(t+1) = m_i(t) + h_{ci}(t)\alpha(t)[x(t) - m_i(t)] \quad \forall i \text{ tel que } d_v(c,i) \leq N_c(t).$$

Le coefficient d'adaptation (ou facteur d'adaptation) $\alpha(t)$, tel que $0 < \alpha(t) < 1$, est strictement décroissant durant l'apprentissage.

$N_c(t)$ définit le niveau de voisinage. Il représente la largeur du voisinage d'adaptation pour le neurone gagnant d'indice c à l'itération de rang t (cf. figure 2.9a).

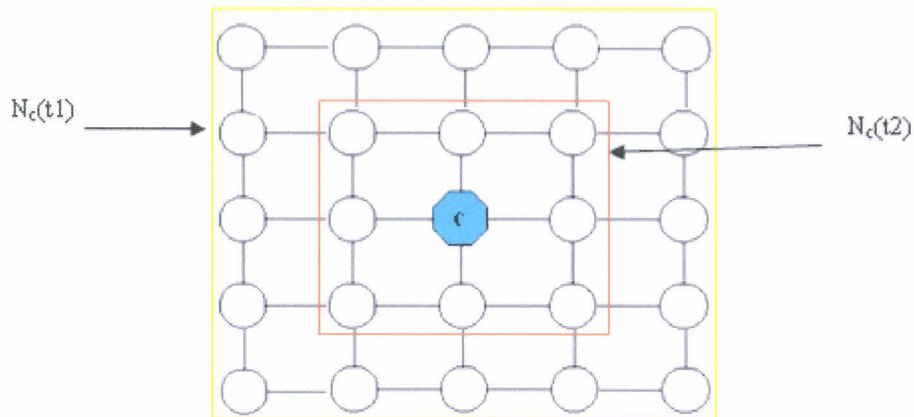
On appelle voisinage d'adaptation l'ensemble des neurones qui participent à l'adaptation. A une itération de rang t , tous les neurones d'indice i dont la distance topologique inter-neurones $d_v(c,i)$ est inférieure ou égale à $N_c(t)$, appartiennent au voisinage d'adaptation du neurone gagnant. $N_c(t)$ doit décroître de manière progressive au cours de l'apprentissage (cf. figure 2.9a).

La fonction $h_{ci}(t)$, appelée fonction de voisinage (ou noyau d'activation) permet de définir l'amplitude de l'adaptation pour chaque neurone d'indice i en fonction de la distance inter-neurones $d_v(c,i)$. Pour garantir la convergence de l'algorithme, il est nécessaire que la fonction $h_{ci}(t)$ tende vers 0 quand t tend vers l'infini.

Dans la littérature, on retrouve généralement deux types de fonctions de voisinage. La plus simple est la fonction constante sur l'ensemble du voisinage d'adaptation (cf. figure 2.9b).

La fonction de voisinage constante $h_{ci}(t)$ est définie comme suit : $h_{ci}(t)=1$ si $d_v(c,i) \leq N_c(t)$ et $h_{ci}(t)=0$ si $d_v(c,i) > N_c(t)$.

L'autre type de fonction de voisinage utilisée est le noyau gaussien (cf. figure 2.9c). Cette fonction est de la forme : $h_{ci}(t) = \exp\left(-\frac{d_v(c,i)^2}{2\sigma^2(t)}\right)$ si $d_v(c,i) \leq N_c(t)$ et $h_{ci}(t)=0$ si $d_v(c,i) > N_c(t)$. $\sigma(t)$ est une fonction monotone, décroissante qui définit la largeur du noyau gaussien.



a- diminution du voisinage au cours du temps avec $t1 < t2$

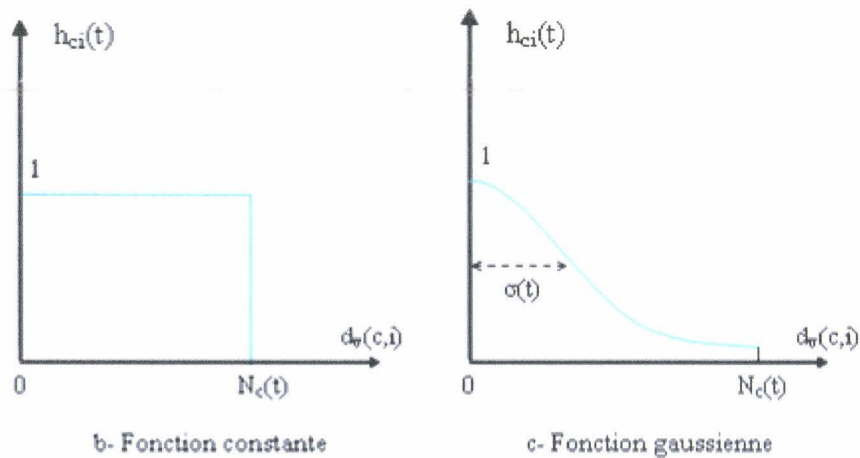


Figure 2.9 Evolution du niveau de voisinage d'adaptation au cours de l'apprentissage (a) et fonctions de voisinage (b et c)

4.2.3. Phases successives d'ordres global et local

Au cours du processus d'apprentissage, Kohonen distingue 2 phases :

- Une organisation globale du réseau : phase d'ordre global
- Un renforcement de l'organisation de manière plus localisée : phase d'ordre local

La notion d'ordre est établie par rapport à la régularité et à l'organisation du réseau durant l'apprentissage. Au début, on assiste à une structuration globale du réseau qui permet de représenter la distribution globale des données dans l'espace d'entrée. Durant cette phase, une majorité des neurones appartenant à la carte sont adaptés à chaque nouvelle présentation d'une donnée en entrée du réseau. Le réseau ainsi sollicité subit dans son ensemble d'importantes

fluctuations de position, tendant à répondre de façon homogène à l'ensemble des données présentées : c'est la phase d'ordre global. Puis, au fur et à mesure que l'apprentissage progresse, que le voisinage d'adaptation diminue, les neurones tendent à se structurer autour de zones plus précises de l'espace d'entrée. Cette évolution marque le début d'une spécialisation des neurones par groupes de plus en plus réduits vis-à-vis de zones particulières de l'espace d'entrée. Durant ce second temps d'adaptation lié à un apprentissage moins étendu, la carte tend à représenter les distributions des données de l'espace d'entrée de manière beaucoup plus locale : c'est la phase d'ordre local. Les phases d'ordre global et local correspondent à des durées d'apprentissage respectives qui ne peuvent pas être déterminées de façon précise car la frontière entre ces deux phases ne peut être fixée durant l'apprentissage que de façon très subjective.

4.2.4. Influence des paramètres

Il faut noter que l'algorithme est relativement sensible à l'ajustement des paramètres $N_c(t)$ et $\alpha(t)$. Par exemple, si le niveau du voisinage d'adaptation à l'itération initiale de rang $t = 0$ est trop faible ou si le coefficient d'adaptation diminue trop rapidement, le réseau n'a pas le temps de se structurer de manière convenable, provoquant des défauts d'adaptation (cf. figure 2.10). De façon générale, si la phase d'ordre global n'est pas suffisante, en terme de nombre de pas d'apprentissage, la représentation de l'espace d'entrée ne peut pas être garantie car une mauvaise adaptation provoque implicitement une mauvaise représentation des données. Au début de l'apprentissage, le niveau du voisinage d'adaptation doit donc être relativement grand, afin de permettre une organisation globale des neurones. Il en va de même pour le coefficient d'apprentissage dont la valeur initiale ne doit pas être trop faible. L'évolution de ces paramètres au cours de l'apprentissage doit par ailleurs rester relativement lente. Une évolution trop rapide pourrait en effet provoquer des distorsions de la carte, sortes de replis de la carte sur elle-même (cf. figure 2.10). La qualité de la représentation des données dans l'espace d'entrée par la carte est également tributaire du nombre de pas d'apprentissage durant la phase d'ordre local. Il est impératif en effet que cette phase soit suffisamment longue pour permettre à la carte de représenter avec suffisamment de précision les données dans l'espace d'entrée.

4.2.5. Durée de l'apprentissage

En ce qui concerne la phase d'apprentissage, l'expérimentation a permis à Kohonen de mettre en place une règle heuristique qui donne de bons résultats : *pour une bonne représentation des données présentées en entrée du réseau, le nombre de pas d'apprentissage*

total doit être au moins égal à 500 fois le nombre de neurones dans la carte [Koh02]. Néanmoins, il n'existe pas vraiment de critère d'arrêt objectif, l'apprentissage se termine généralement quand les paramètres $N_c(t)$ et $\alpha(t)$ atteignent des valeurs faibles et que les vecteurs-poids n'évoluent pratiquement plus.

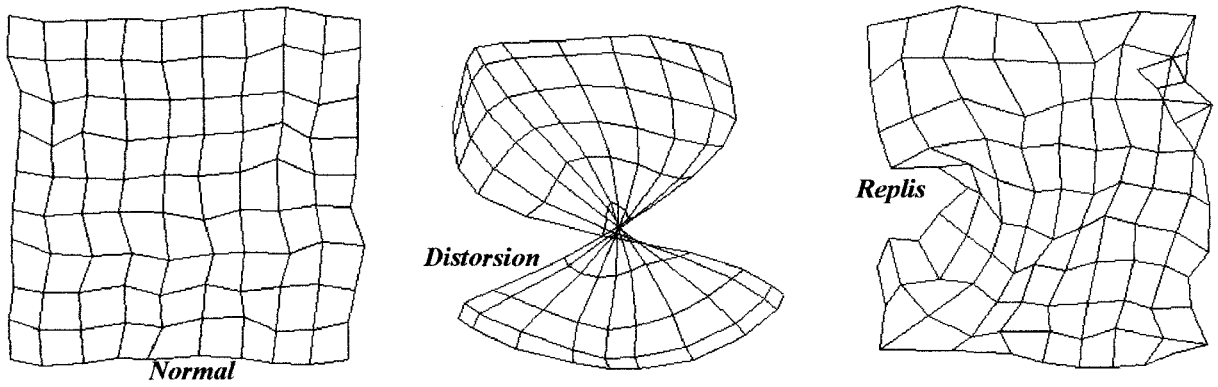


Figure 2.10 Défauts d'adaptation

4.3. Algorithme de Kohonen

Formulation de l'algorithme de Kohonen appliqué à une carte 2D rectangulaire :

ALGORITHME

- 1- A l'itération de rang $t=0$, initialisation du nombre de lignes et de colonnes de la carte.
- 2- Initialisation aléatoire des vecteurs-poids m_i , $i=1,2,\dots,M$.
- 3- Présentation d'une donnée $x(t)$ en entrée du réseau .
- 4- Détermination du neurone vainqueur $c = \underset{i}{\operatorname{argmin}} \left\{ \|x(t) - m_i(t)\| \right\}$.
- 5- Mise à jour des poids à l'aide de la règle suivante :

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) \leq N_c(t) \\ m_i(t); & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

6- Mise à jour des paramètres d'adaptation ($N_c(t)$, $\alpha(t)$). Incrémentation du nombre de pas d'apprentissage $t=t+1$. Répétition de la procédure de 3 à 5 jusqu'à ce qu'un nombre maximal d'itérations T_{Max} soit atteint, ou que l'utilisateur juge visuellement que l'adaptation est terminée.

4.4. Exemples d'application

- **Application d'une carte 1D au problème du voyageur de commerce**

L'architecture du réseau 1D a la forme d'une structure filaire en boucle ouverte ou fermée. Il est constitué d'une seule couche de neurones où chaque neurone est lié à ses deux voisins, sauf pour le cas des neurones situés aux extrémités d'une structure à boucle ouverte qui ne possède qu'un seul voisin. Nous présentons un exemple d'application de cette architecture pour résoudre le célèbre problème du « voyageur de commerce ». Nous considérons un ensemble de V villes, représentées chacune par un point de coordonnées V_x, V_y dans un plan. Il s'agit de trouver le plus court chemin fermé passant une fois et une seule par chacune des villes. La difficulté du problème vient de l'explosion combinatoire du nombre de chemins à explorer lorsque l'on accroît le nombre de villes à visiter. Si on teste les $\frac{1}{2}(V-1)!$ circuits possibles, on est assuré de découvrir le chemin le plus court. Mais la complexité de cette méthode de recherche exhaustive augmente de manière combinatoire avec le nombre de villes V . Pour $V=30$, le temps de calcul a été estimé comme supérieur à l'âge de l'univers [Hér94].

La convergence de l'algorithme vers une solution est principalement liée au voisinage d'adaptation durant l'apprentissage. Un vecteur d'entrée représentant une ville de coordonnées V_x, V_y est choisi au hasard parmi les villes. Le processus compétitif prend place et un neurone est déclaré vainqueur. Ce neurone, ainsi que les neurones appartenant à son voisinage, sont adaptés pour se rapprocher de la ville choisie en entrée. L'apprentissage du réseau, qui implique la notion de voisinage dans le mécanisme d'adaptation, permet d'avoir une représentation globale et locale de la distribution de l'ensemble des villes présentées en entrée du réseau et de les représenter par des neurones voisins dans le réseau. Ainsi, au fil de l'apprentissage et de proche en proche, chaque groupe de neurones voisins se spécialise par rapport à un groupe de villes voisines.

La figure 2.11, illustre ce problème et sa résolution en utilisant un réseau de type SOCNet [Bie99] qui est une carte auto-organisatrice filaire bouclée («Self-Organizing Circular Network»). L'espace d'entrée représente les différentes villes que doit parcourir le voyageur de

commerce (cf. figure 2.11a). Le réseau est initialisé aléatoirement avant que l'apprentissage ne débute (cf. figure 2.11b). Durant l'apprentissage, le réseau se déplie et converge vers une configuration d'architecture représentant un tour possible des villes (cf. figure 2.11d). Ce chemin représente une solution au problème mais il n'y a aucune assurance sur l'optimalité de la longueur totale du chemin trouvé, puisqu'il n'y a pas de critère d'optimisation explicite dans le processus d'adaptation du réseau.

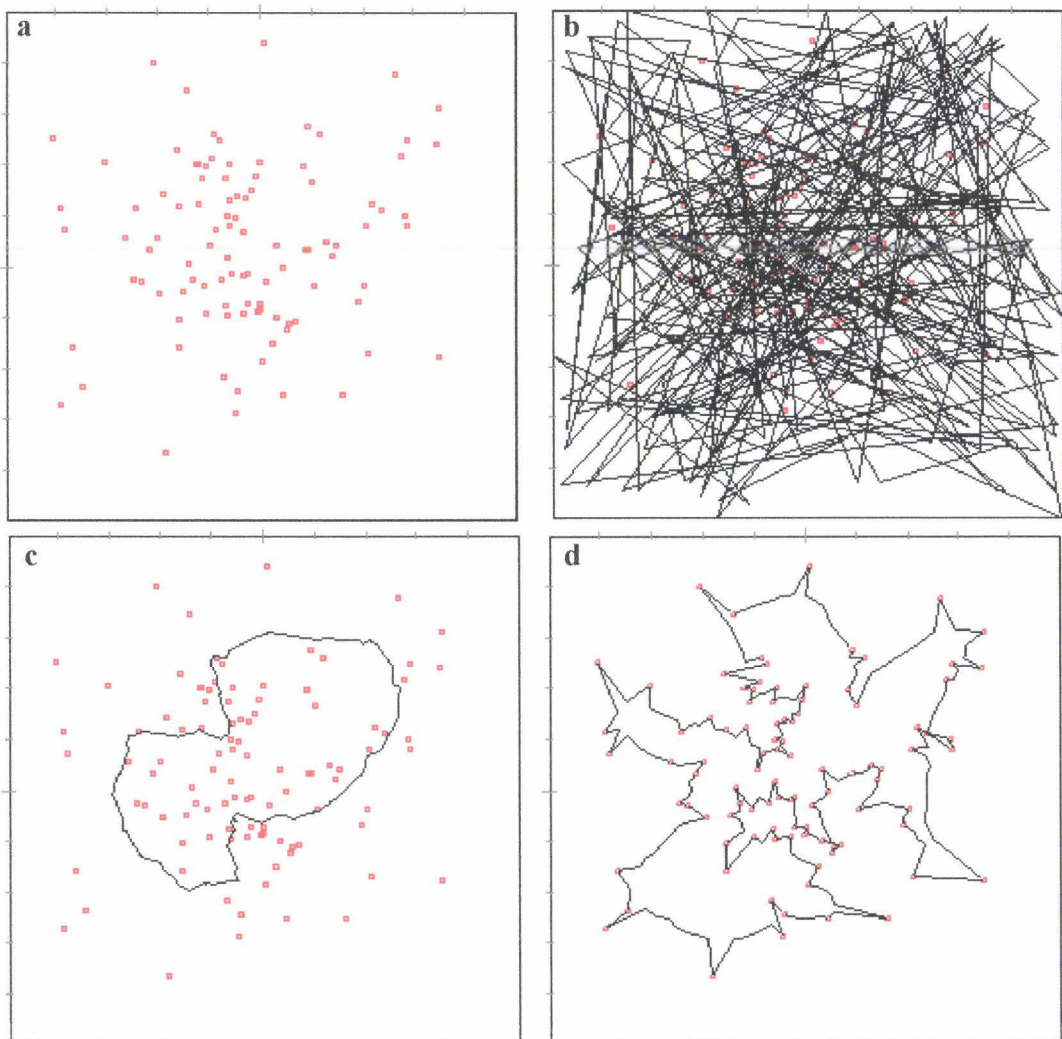


Figure 2.11 Résolution du problème du voyageur du commerce à l'aide d'un réseau de type SOCNet.

a – représentation de 100 villes sur un plan,

b – initialisation aléatoire des 250 neurones du réseau,

c- configuration de la carte après 1000 itérations,

d – configuration finale de la carte après 3000 itérations.

- **Maillage d'un morceau de plan à l'aide d'une carte 2D**

Dans un second exemple, on considère un espace d'entrée délimité par un carré défini dans le plan (x,y) par $-1 < x < +1$ et $-1 < y < +1$. Des points de coordonnées (x,y) sont répartis de façon aléatoire suivant une distribution uniforme à l'intérieur de ce plan. L'ensemble de ces points constitue la base d'apprentissage et on désire pouvoir représenter la distribution de ces points dans le plan par une carte de Kohonen 2D, les vecteurs-poids des neurones constituant les nœuds du maillage (cf. figure 2.12).

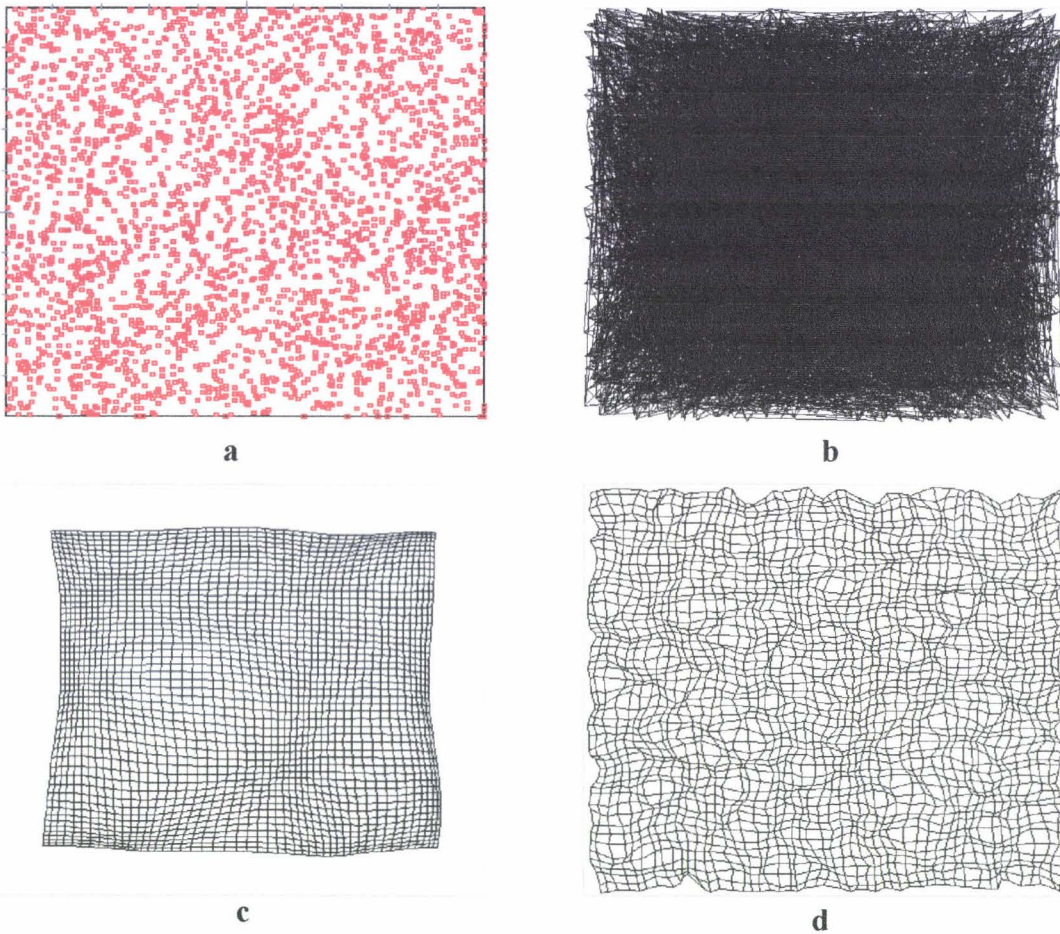


Figure 2.12 Adaptation d'une carte rectangulaire 2D (50x50 neurones).

a- nuage de 3000 points,

b- initialisation aléatoire des neurones,

c- la carte après 1000 itérations,

d- la carte après 10000 itérations.

Après initialisation de la carte 2D de façon aléatoire à l'intérieur du périmètre du plan (cf. figure 2.12b), nous constatons que l'organisation progressive de la carte tout au long de l'apprentissage s'effectue de façon harmonieuse (cf. figure 2.12c) pour aboutir en fin d'apprentissage à un ensemble représentant de façon cohérente l'uniformité de distribution des points dans le plan tout en respectant la topologie de cette distribution puisque la carte prend elle-même la forme d'une surface carrée (cf. figure 2.12d).

- **Reconstruction de surfaces 3D : limites du modèle 2D**

La carte 2D, en forme de grille, est également très efficace pour la reconstruction de surfaces 3D. Pour illustrer cette propriété des cartes 2D, nous avons réalisé le maillage de la surface d'une carrosserie de voiture à partir d'un nuage des points 3D acquis à la surface d'un véhicule (cf. figure 2.13a). Le résultat de l'adaptation est donné sur la figure 2.13b qui montre bien que la carte a su s'adapter aux formes de la carrosserie.

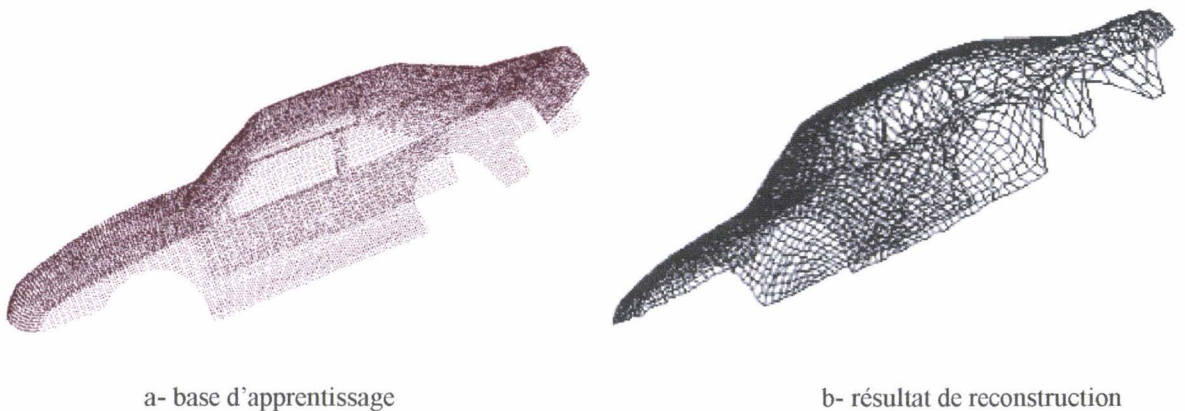


Figure 2.13 Reconstruction de la carrosserie d'une voiture par carte 2D

Néanmoins la reconstruction de surfaces 3D par une carte 2D peut montrer certaines lacunes comme dans le cas de la représentation d'une sphère (cf. figure 2.14). Dans ce cas de figure, il apparaît des zones qui ne sont pas mal ou même ne sont pas représentées par la carte 2D. Evidemment, il n'est pas possible de représenter une surface 3D fermée par une surface 3D ouverte. Ce constat met en évidence la nécessité d'adapter l'architecture de la carte dans le cas particulier où l'on cherche à représenter des surfaces fermées d'objets 3D.

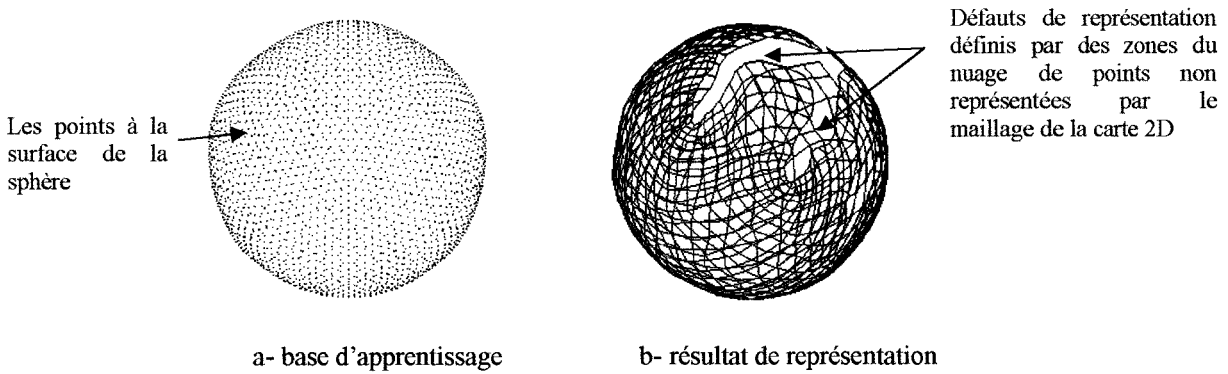


Figure 2.14 Reconstruction d'une sphère par carte 2D

Malgré les défauts de représentation finale, les exemples que nous avons présentés montrent la capacité de représentation par maillage de surfaces 3D à l'aide de cartes auto-organisatrices. Pour fournir des résultats beaucoup plus probants, il est cependant nécessaire d'utiliser une forme d'architecture neuronale différente, mieux adaptée à la topologie particulière engendrée par les surfaces 3D fermées. Nous présentons dans la suite du mémoire l'ensemble des éléments relatifs à cette démarche.

5. Convergence de l'algorithme

La carte auto-organisatrice de Kohonen qui est simple dans son concept, est néanmoins efficace. Mais son comportement s'avère particulièrement difficile à analyser de façon formelle. Dans le cas des cartes monodimensionnelles et pour des densités de distribution uniformes, la convergence de l'algorithme a été partiellement prouvée par Kohonen [Koh82a]. Cottrell et Fort [Cot87] ont ensuite fourni les premières preuves complètes d'auto-organisation et les propriétés de convergence pour une distribution uniforme des entrées et un voisinage simple, toujours dans le cas unidimensionnel. Bouton et Pagès [Bou93] et Erwin [Erw92a, b] ont généralisé ces résultats à une large classe de distributions des entrées et à une fonction plus générale de voisinage.

Les principaux résultats théoriques connus sont réunis dans le papier de synthèse de Cottrell, Fort et Pagès [Cot94]. Pour des dimensions plus élevées, les résultats sont seulement partiels. Deux difficultés majeures expliquent cette situation. La première concerne la difficulté d'exprimer la notion d'ordre dans un réseau maillé à plus d'une dimension qui serait stable pour

l'algorithme : les configurations de grille que Lo propose en 1991 et 1993 [Lo91], [Lo93] sont effectivement instables. Deuxièmement, Erwin et al. en 1992 [Erw92a, b] ont prouvé qu'il est impossible d'associer une fonction potentielle globale de décroissance à l'algorithme si la distribution de probabilité des entrées est continue.

Ritter et al. en 1986 et 1988 [Rit86], [Rit88] ont apporté un éclairage sur l'état stationnaire dans le cas multidimensionnel, mais ils ne se sont intéressés qu'à la phase d'ordre local et ils ne prouvent pas l'existence d'un état stationnaire. Plus récemment, Fort et Pagès donnent certains résultats dans des dimensions plus élevées, mais ces résultats sont très partiels [For93].

Par ailleurs, certains résultats théoriques sont valables dans différents contextes. En 1992, Thiran et Hasler [Thi92] donnent une preuve de l'auto-organisation en dimension 1 pour une large classe de fonctions de voisinage pour des poids synaptiques quantifiés et des entrées discrètes. En 1992, Ritter et al. [Rit92] introduisent une fonction potentielle quand les entrées n'appartiennent qu'à un ensemble fini de valeurs discrètes. L'algorithme de Kohonen ne représentait qu'un principe de descente de gradient stochastique sur cette même fonction, du moins pour chaque valeur constante de la fonction de voisinage. Finalement, en 1992, Pagès [Pag92] fournit une étude complète du cas où il n'existe pas de notion de voisinage.

6. Conclusion

Dans ce chapitre nous avons fait une présentation des cartes auto-organisatrices, de leurs origines biologique jusqu'aux présentations de quelques exemples pratiques d'application (résolution du problème de voyageur de commerce, maillage de surfaces 2D). Dans ce contexte, nous avons également présenté leurs principales caractéristiques.

Dans le cadre de cette présentation, nous avons mis en exergue le besoin de définir de nouvelles architectures de cartes auto-organisatrices pour la reconstruction de surfaces 3D.

Dans les prochains chapitres, nous allons présenter la démarche que nous avons suivie pour la reconstruction de surfaces 3D à l'aide de cartes auto-organisatrices appelées 3D-SOM.

Les réseaux 3D-SOM

1. Introduction

Dans le chapitre 2, nous avons présenté les principes généraux des cartes auto-organisatrices de Kohonen : architecture, apprentissage, paramètres d'adaptation, représentation des données, etc.. Dans le cadre de la reconstruction et de la représentation de formes d'objets 3D à partir de nuages de points, nous y avons également mis en évidence les limitations de l'architecture de base (carte 2D) pour des applications dédiées à la reconstruction de surfaces 3D fermées. Dans ce chapitre, nous présentons une architecture mieux adaptée à cette problématique que nous appellerons 3D-SOM (« 3D-Self-Organizing Map »). Le réseau 3D-SOM est une architecture neuronale, inspirée des cartes auto-organisatrices de Kohonen, dédiée à la reconstruction et représentation de formes 3D à partir de nuages de points représentatifs.

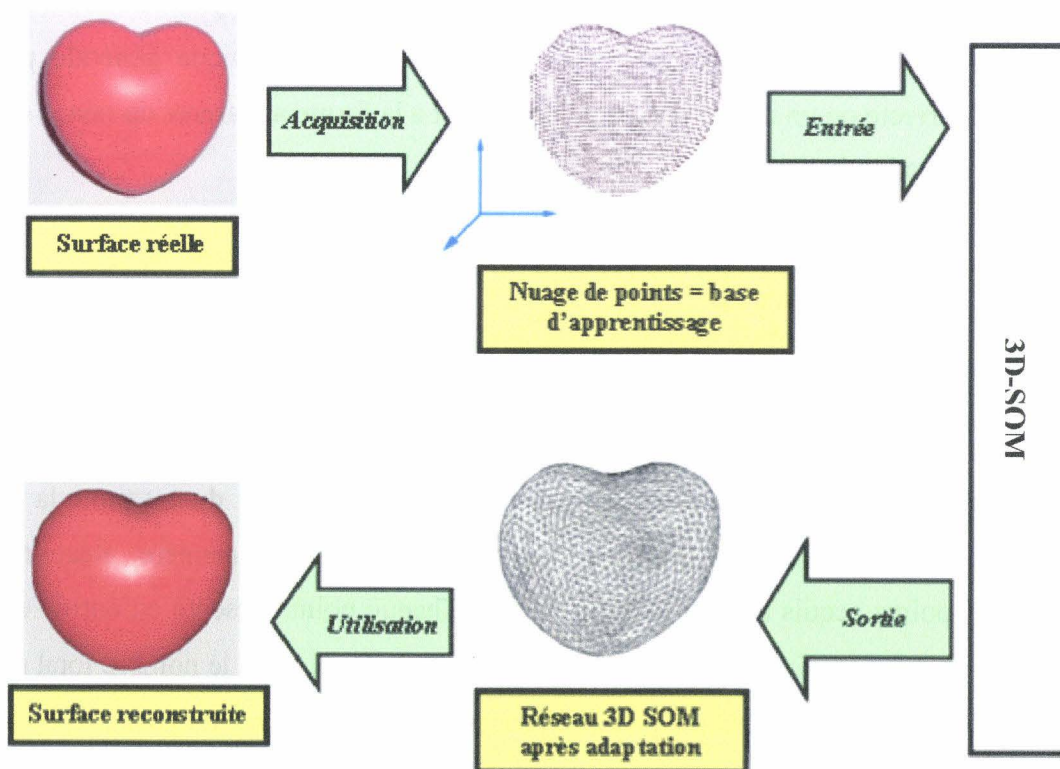


Figure 3.1 Représentation schématique des étapes liées à la chaîne de reconstruction d'une surface 3D à partir d'un objet réel

La figure 3.1 illustre l'ensemble du processus de reconstruction qui consiste d'abord en l'acquisition d'un nuage de points situés à la surface de l'objet à représenter et repérés par leurs 3 coordonnées spatiales. L'ensemble de ces points d'échantillonnage constitue la base d'apprentissage du réseau 3D-SOM. Après adaptation, la dernière phase du processus consiste en l'utilisation effective du modèle maillé de la surface pour une représentation en rendu réaliste de l'objet reconstruit.

Notons enfin qu'il n'est pas simple d'exprimer de façon quantitative la qualité ou l'exactitude de la reconstruction obtenue et il n'existe pas de solution triviale à ce problème. Par ailleurs les réseaux 3D-SOM réalisent une adaptation non déterministe de leur architecture neuronale par rapport à un nuage de points présenté en entrée du réseau. De fait les 3D-SOM sont conçus avant tout pour permettre une interprétation visuelle de la répartition et de la correspondance topologique entre les données dans l'espace de représentation : cette propriété a déjà été mentionnée au chapitre 2 pour la représentation et l'interprétation de données à l'aide des cartes de Kohonen dont les 3D-SOM représentent une extension particulière. Dans le cadre de notre travail, nous nous baserons donc exclusivement sur des aspects plus ou moins subjectifs pour apprécier la qualité d'une reconstruction, tels l'aspect d'esthétique générale pour une reconstruction correcte ou un défaut d'aspect par rapport à la forme apparente du nuage de points pour une reconstruction imparfaite.

2. Architecture des réseaux 3D-SOM

2.1. Introduction

A l'image des cartes de Kohonen, le réseau 3D-SOM est composé de deux couches : une couche d'entrée et une couche compétitive. La couche d'entrée reçoit, durant toute la durée de l'apprentissage, les coordonnées spatiales des points choisis de manière aléatoire parmi l'ensemble des points acquis à la surface de l'objet. Chaque point présenté en entrée est défini par un vecteur x_j avec $x_j = [x_{j1}, x_{j2}, x_{j3}]^T \in \mathcal{R}^3$, avec $j=1,2,\dots,N$, N étant le nombre total de points dans la base d'apprentissage. Le principe de ce processus est illustré sur la figure 3.2.

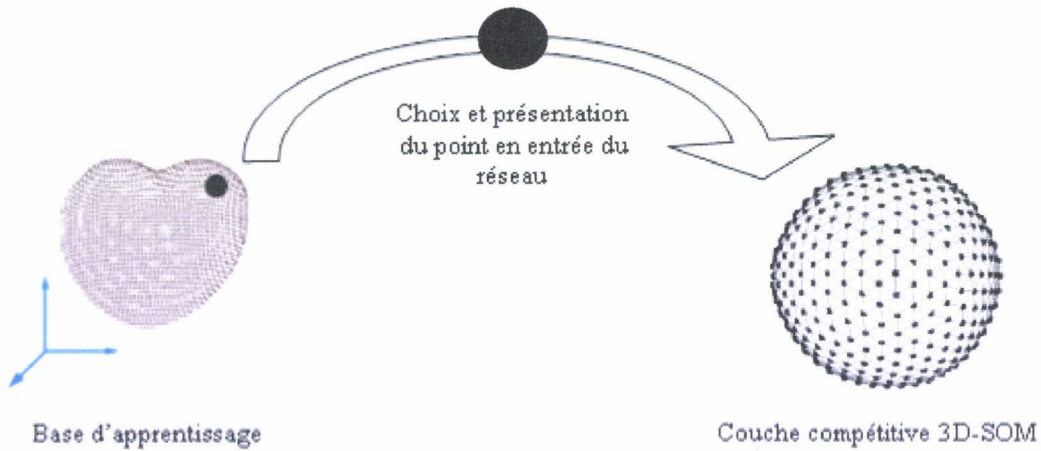


Figure 3.2 Choix aléatoire d'un point dans la base d'apprentissage et présentation de ce point au réseau 3D-SOM

La structure du réseau représente une architecture 3D fermée qui peut prendre différentes formes : icosaèdre, cube, tétraèdre, tore, structure synthétique, etc. Pour décrire cette structure à partir de la topologie des relations de voisinage entre les neurones du réseau, nous avons choisi le maillage triangulaire. Le triangle représente en effet l'élément de forme planaire le plus simple et il est aisé de construire diverses formes 3D à partir de la combinaison de ces éléments de base (cf. figure 3.3). Dans une telle construction, les arêtes des triangles représentent naturellement les relations de voisinage entre les neurones, tandis que les sommets des triangles représentent les neurones eux-mêmes. Chaque neurone d'indice i est repéré par son vecteur-poids $m_i = [m_{i1}, m_{i2}, m_{i3}]^T \in \mathbb{R}^3$, avec $i=1,2,\dots,M$, M étant le nombre total de neurones dans le réseau. Les valeurs des composantes du vecteur-poids représentent la position du neurone dans l'espace de représentation tridimensionnel.

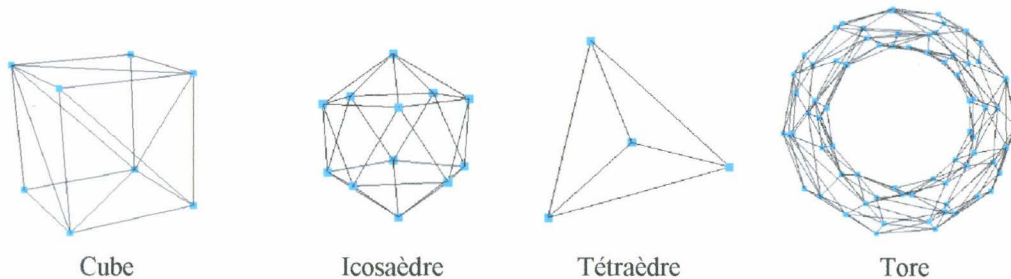


Figure 3.3 Exemples d'architectures de 3D-SOM avec maillage triangulaire

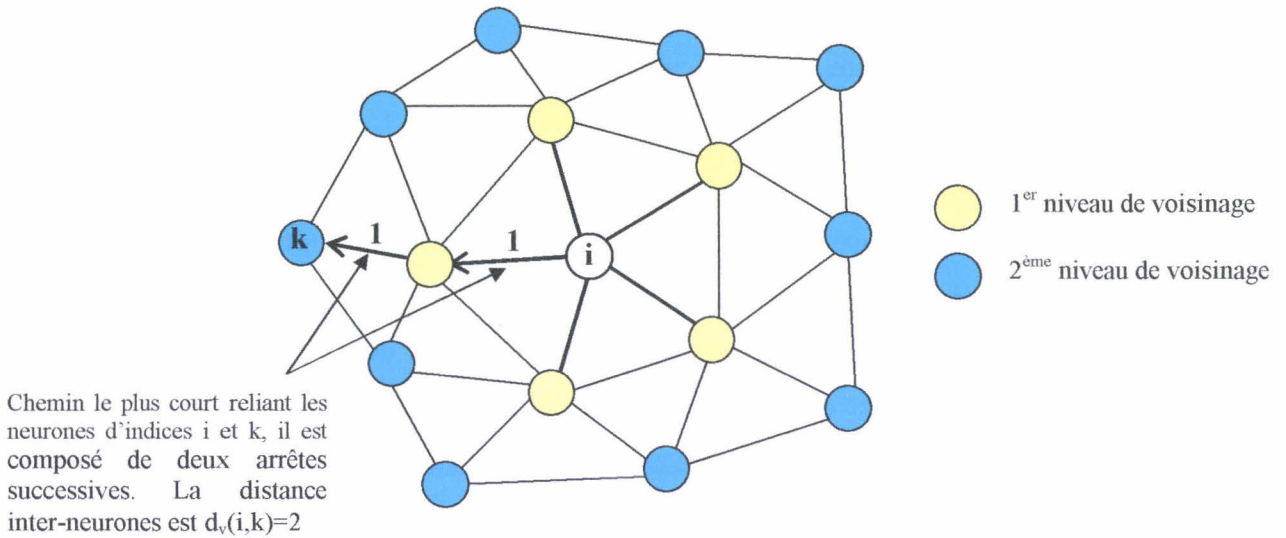
Notons que les cartes graphiques modernes que l'on trouve sur la plupart des ordinateurs actuels permettent une manipulation directe des éléments graphiques de surface que sont les triangles, appelés aussi facettes, pour réaliser un affichage très rapide des objets 3D décrits à l'aide de maillages triangulaires.

2.2. Généralisation de l'architecture des cartes auto-organisatrices

La représentation topologique des relations de voisinage entre les neurones du réseau 3D-SOM est faite sous forme de graphe. Cette représentation offre l'avantage de permettre un parcours de l'ensemble du réseau en suivant les liens de voisinage entre neurones successifs suivant des méthodologies déjà bien établies [Die97]. Ainsi, dans notre cas, l'implantation du graphe se fera sous la forme d'un graphe d'adjacence, qui offre la possibilité de représenter l'architecture topologique du réseau 3D-SOM sans utiliser énormément de ressources machine.

Les nœuds du graphe représentent les neurones du réseau. Les arêtes représentent les liaisons de voisinage entre les neurones et permettent de définir la notion de distance topologique inter-neurones. La distance inter-neurones à l'intérieur du graphe est définie comme la longueur du chemin le plus court entre deux neurones. Les neurones consécutifs constituant ce chemin sont nécessairement des neurones directement voisins reliés par une seule arête. La longueur d'un chemin entre deux neurones est définie par le nombre d'arêtes les séparant. La figure 3.4 illustre cette notion de longueur de chemin.

La notion de distance inter-neurones permet d'établir les différents niveaux de voisinage possibles pour chaque neurone par rapport à ses voisins immédiats ou lointains. Dans l'exemple précédant, le neurone d'indice k appartient au deuxième niveau de voisinage par rapport au neurone d'indice i .



*Figure 3.4 Principe de l'organisation topologique
des neurones à l'intérieur des 3D-SOM*

La notion de premier niveau de voisinage pour un neurone d'indice i est définie par les neurones qui ont un lien direct avec lui, c'est-à-dire séparés par une seule arrête avec une distance inter-neurones de valeur 1. Le deuxième niveau de voisinage d'un neurone d'indice i est composé des neurones dont la distance inter-neurones les séparant est de 2, etc. (cf. figure 3.4).

La représentation des liens topologiques entre neurones sous forme de graphe d'adjacence offre l'avantage d'une généralisation de la notion d'architecture topologique des cartes auto-organisatrices, en partant du fait que toute forme représentée par un maillage triangulaire peut être utilisée comme architecture topologique de la couche compétitive d'une carte auto-organisatrice.

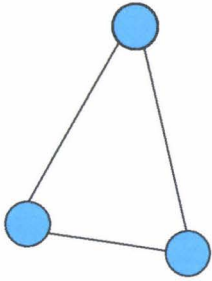
2.3. Subdivision totale

La représentation d'objets 3D acquis avec une densité de points importante nécessite des architectures suffisamment denses en nombre de neurones pour permettre une représentation satisfaisante. Dans ce cas, les architectures de base présentées sur la figure 3.3 ne sont évidemment pas adaptées car le nombre total de neurones reste beaucoup trop faible. Pour répondre à ce problème, il est possible d'augmenter la densité des neurones à l'intérieur du réseau par une méthode de subdivision des mailles.

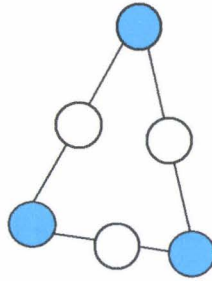
La subdivision consiste à créer 4 nouveaux triangles à partir de chaque triangle déjà existant, ce qui implique l'apparition de 3 nouveaux neurones. Le détail de cette méthode est illustré sur la figure 3.5a.

Chaque nouveau neurone est généré au milieu de chaque arête liant deux neurones existants. Le vecteur-poids du nouveau neurone est initialisé avec les coordonnées 3D du milieu de l'arête où il est créé. Pour chaque triangle initial composé de 3 neurones, les 3 nouveaux neurones qui se positionnent sur ses bords forment les sommets des arêtes de 4 nouveaux triangles. Cette méthode de subdivision est appelée subdivision totale car elle est appliquée à la totalité du maillage du réseau. La figure 3.5b illustre ce processus de subdivision totale, en mettant en relief l'évolution du maillage au fur et à mesure que se font les étapes de création des neurones, puis des triangles, dans le maillage.

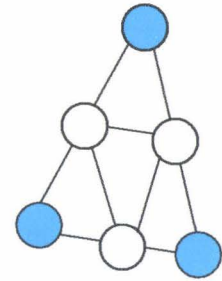
L'illustration de la figure 3.5c présente l'exemple d'un tétraèdre subissant deux étapes successives de subdivision totale. Partant d'une structure initiale composée de 4 neurones formant 4 triangles, nous retrouvons à la fin de la deuxième étape de subdivision une architecture composée de 34 neurones et 48 triangles.



Triangle initial composé de trois neurones, avant subdivision

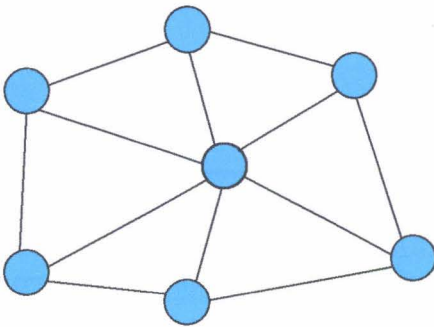


Création de 3 nouveaux neurones au milieu des arêtes

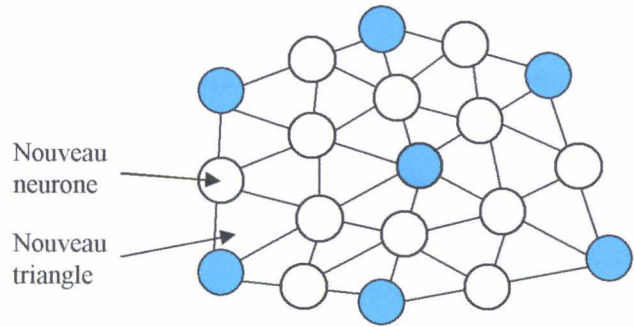


Création de 3 nouvelles arêtes entre les nouveaux neurones et apparition de 4 nouveaux triangles

a- illustration du processus de subdivision sur un triangle

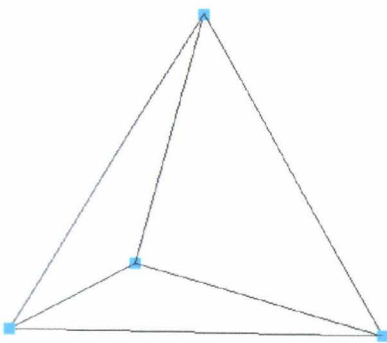


Avant subdivision

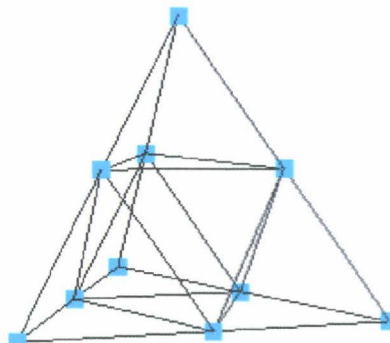


Après subdivision

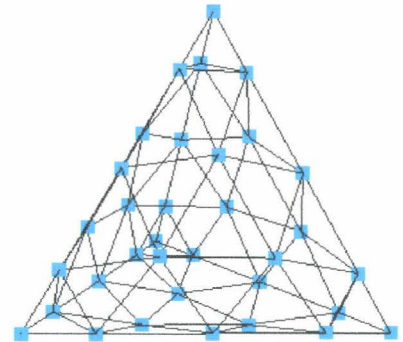
b- illustration du processus de subdivision totale



Tétraèdre
4 neurones
4 triangles



Tétraèdre après une 1er étape de subdivision totale
10 neurones
16 triangles



Tétraèdre après une 2ème étape de subdivision totale
34 neurones
48 triangles

c- illustration de la subdivision totale sur un tétraèdre

Figure 3.5 Illustration du principe de la méthode de subdivision totale

3. Apprentissage compétitif

Durant la phase d'adaptation, l'architecture du réseau doit pouvoir se structurer et s'organiser de façon à permettre la représentation de la forme du nuage de points présenté en entrée. Les réseaux 3D-SOM suivent les grands principes de l'apprentissage compétitif qui, de façon très générale, implique de façon répétitive une compétition entre les neurones du réseau dans le but d'élire un neurone vainqueur. La procédure d'apprentissage s'applique ensuite à l'ensemble des neurones appartenant au voisinage d'adaptation de ce neurone vainqueur.

3.1. Apprentissage compétitif simple

3.2.1. Propriétés de l'apprentissage compétitif simple

Le principe de cet apprentissage est directement issu des techniques d'apprentissage compétitif liées aux cartes auto-organisatrices de Kohonen [Koh01]. L'apprentissage compétitif simple consiste, à chaque présentation d'un point en entrée du réseau, à sélectionner le neurone qui lui en est le plus proche, ainsi que les neurones voisins, puis à modifier les poids de ces neurones dans le but de les rapprocher du point présenté en entrée. Ce mécanisme auto-adaptatif permet à des neurones topologiquement voisins dans le réseau de devenir sensibles à des entrées voisines au sens d'une distance mesurée dans l'espace de représentation. L'itération de ce processus se poursuit tout au long de la phase d'apprentissage [Koh82].

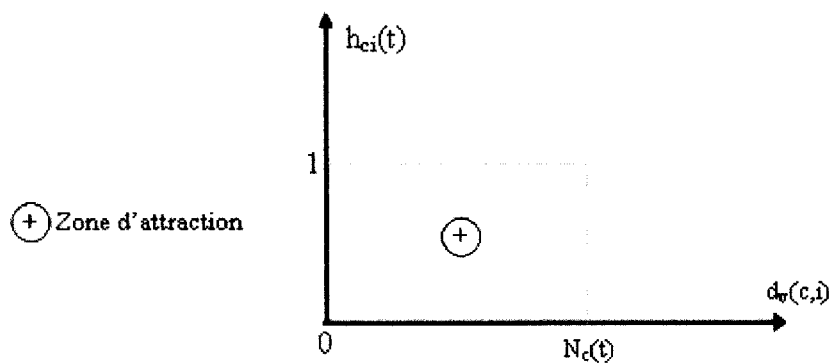


Figure 3.6 Fonction de voisinage constante pour l'apprentissage compétitif simple

La fonction de voisinage utilisée pour les neurones qui subissent l'adaptation est la fonction constante (cf. figure 3.6). Ce choix est motivé pour des raisons de simplicité. En effet, pour une fonction constante, la fonction de voisinage est définie comme suit :

$$h_{ci}(t) = \begin{cases} 1; & \text{si } d_v(c,i) \leq N_c(t) \\ 0; & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

$N_c(t)$ représente le niveau du voisinage d'adaptation pour le neurone gagnant d'indice c à l'instant de rang t , où t est un entier qui représente le rang du pas d'apprentissage en cours.

3.2.2. Algorithme pour l'apprentissage compétitif simple

- 1- Initialisation des vecteurs-poids m_i pour $t = 0$.
- 2- A l'itération de rang t , un point x_j , qu'on notera $x(t)$, est choisi de manière aléatoire dans l'espace d'entrée. $x(t)$ est présenté à l'entrée du réseau pour être comparé à tous les vecteurs-poids $m_i(t)$ en utilisant la distance euclidienne $d_e(x(t), m_i(t))$. Le neurone pour lequel la distance $d_e(x(t), m_i(t))$ est la plus petite est déclaré vainqueur : son indice est noté c et son vecteur-poids $m_c(t)$, avec $c = \underset{i}{\operatorname{Argmin}} \{ d_e(x(t), m_i(t)) \}$.
- 3- Le vecteur-poids du neurone vainqueur ainsi que ceux des neurones appartenant au voisinage d'adaptation sont adaptés de manière à se rapprocher du point $x(t)$ présenté en entrée. Ainsi, si $m_c(t)$ est le vecteur poids du neurone vainqueur pour l'entrée $x(t)$, la règle d'apprentissage dans son voisinage est définie comme suit :

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) \leq N_c(t) \\ m_i(t); & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

Le coefficient d'apprentissage $\alpha(t)$ est défini dans l'intervalle $[0,1]$. $\alpha(t)$ et $N_c(t)$ décroissent graduellement tout au long de l'apprentissage. La loi de décroissance pour $\alpha(t)$ est : $\alpha(t) = \alpha(0)/t^D$. $N_c(t)$ décroît selon la règle : $N_c(t) = N_c(0) \operatorname{DIV} t^d$, où DIV représente la division entière. Les exposants D et d sont définis dans l'intervalle $[0,1]$.

- 4- Si le nombre d'itérations maximal $T_{\max}$ défini par l'utilisateur n'est pas atteint, un retour à l'étape 2 permet de répéter le processus. L'utilisateur peut arrêter à tout instant le processus, s'il estime que l'adaptation est finie avant même que le rang d'itération t atteigne la

valeur maximale T_{Max} . L'apprentissage se termine généralement quand les vecteurs-poids des neurones n'évoluent pratiquement plus. C'est ce critère d'arrêt qui sera pris en compte dans ce qui suit.

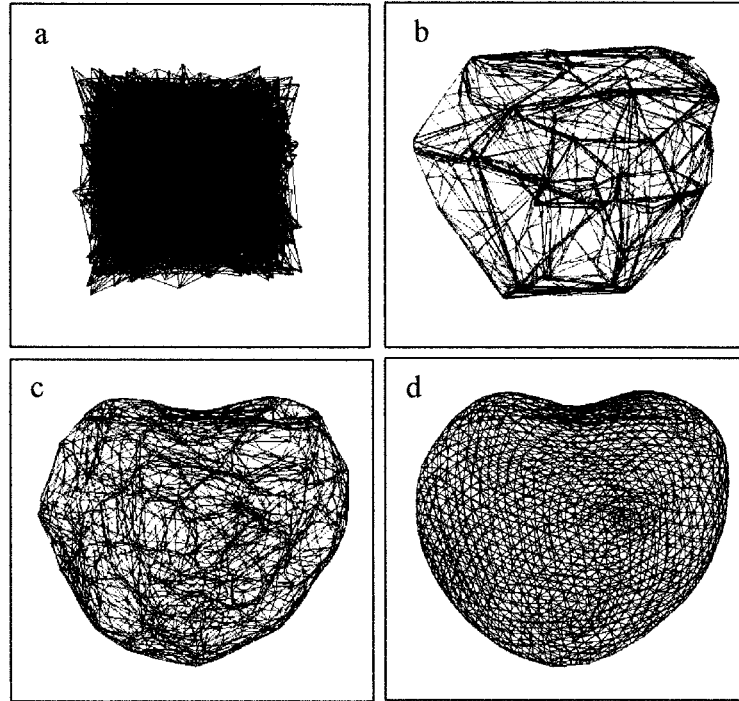


Figure 3.7 Processus d'adaptation d'un réseau 3D-SOM composé de 5122 neurones. La base d'apprentissage est composée de 6402 points représentant la surface d'un cœur.

$$\alpha(0)=0.5, D=0.3, N_c(0)=1000, d=0.6.$$

- a- initialisation aléatoire de tous les vecteurs-poids,*
- b- état du réseau 3D-SOM après 2000 itérations du processus,*
- c- état du réseau 3D-SOM après 13000 itérations du processus,*
- d- adaptation finale après 20000 itérations du processus.*

Nous illustrons le processus d'adaptation d'un réseau 3D-SOM avec apprentissage compétitif simple sur la figure 3.7. Nous y représentons différentes phases d'adaptation d'un réseau dont la structure synthétique composée de 10 triangles est subdivisée préalablement 5 fois. La base d'apprentissage de ce réseau, composée de 5122 neurones, est un nuage de 6402 points acquis à la surface du cœur représenté figure 3.1.

Dans le cas de l'exemple de la figure 3.7, nous pouvons remarquer 2 phases au cours du processus d'adaptation du réseau 3D-SOM : une phase d'ordre global suivie, au fur et à mesure

que l'apprentissage progresse, d'une phase d'ordre local. Ces 2 phases ont été définies par Kohonen dans le cas général de l'adaptation des cartes auto-organisatrices et nous en avons déjà présenté les principales caractéristiques dans le chapitre 2. Les figures 3.7a et 3.7b mettent en évidence l'évolution du réseau au cours de la phase d'ordre global qui a pour but de rendre compte de la distribution globale des points présentés en entrée du réseau. Durant cette première phase, les vecteurs-poids subissent d'importantes variations et ne sont pas encore tout à fait positionnés par rapport à des régions spécifiques de l'espace d'entrée. Les représentations des positions des neurones dans l'espace d'entrée peuvent ainsi subir de grandes fluctuations d'une itération à l'autre. Ce phénomène résulte notamment des valeurs importantes prises par les paramètres $N_c(0)$ et $\alpha(0)$ à l'itération initiale de rang $t=0$. Dans la seconde phase, dite phase d'ordre local, la majorité des neurones vont, avec leur voisinage, se spécialiser de façon beaucoup plus marquée par rapport à des régions spécifiques de l'espace d'entrée. Les vecteurs-poids subissent une adaptation plus locale avec des variations moins importantes, vu qu'ils sont déjà proches des points présentés en entrée pour lesquels ils sont respectivement élus vainqueurs. Cette phase est dite d'ordre local. C'est à ce stade du processus d'adaptation que le réseau « apprend » les détails de la surface à représenter. Les figures 3.7c et 3.7d illustrent l'évolution de la forme du réseau durant cette seconde phase d'ordre local. Le maillage neuronal résultant (cf. figure 3.7d) représente une approximation de la surface dont est issu le nuage de points en entrée.

3.2. Apprentissage compétitif avec répulsion

Ce mode d'apprentissage s'inspire de mécanismes adaptatifs liés à des structures neuronales présentes dans le cortex cérébral. On y constate l'existence de connexions de natures contraires entre les neurones, excitatrices pour les courtes distances et inhibitrices pour des distances plus longues entre les neurones.

De façon schématique, l'amplitude et la nature de ces interactions entre les neurones en fonction de la distance qui les sépare suit un profil en forme de "chapeau mexicain". De nombreux auteurs ont proposé différents modèles de ce mécanisme adaptatif compétitif. Kohonen a également proposé une version simplifiée de ce principe d'apprentissage compétitif et l'a appliquée aux cartes auto-organisatrices. Nous présentons dans ce paragraphe une approche similaire que nous appliquons aux réseaux 3D-SOM dans le but d'améliorer l'efficacité de l'adaptation.

3.2.1. Inconvénients liés à l'apprentissage compétitif simple

Dans l'apprentissage compétitif simple, partant d'une architecture où les vecteurs-poids des neurones sont initialisés aléatoirement, le voisinage d'adaptation initial doit être très grand, voire total, c'est-à-dire que tous les neurones du réseau vont participer au processus d'apprentissage. Un grand niveau du voisinage d'adaptation permet en effet une organisation globale du réseau cohérente et homogène par rapport à la distribution du nuage des points présentés en entrée. Au début de l'adaptation, pratiquement tous les neurones participent à l'apprentissage des points présentés en entrée du réseau : c'est la phase d'ordre global. Cette phase se révèle généralement assez longue du fait de la nécessité d'obtenir une organisation suffisante du réseau pour atteindre la phase d'ordre local. Kohonen [Koh01] préconise que le niveau du voisinage d'apprentissage $N_c(t)$ reste constant durant les premières itérations d'apprentissage afin de favoriser le déploiement du réseau et éviter ainsi de possibles défauts d'adaptation liés à une mauvaise répartition globale de la carte en début de procédure d'apprentissage.

Au cours de cette première phase, on constate donc beaucoup de déplacements « inutiles » pour les neurones à l'intérieur de l'espace de représentation. En effet, des neurones qui s'adaptent à une zone particulière du nuage de points en début d'apprentissage ne représenteront probablement pas cette même zone en fin d'apprentissage.

3.2.2. Propriétés de l'apprentissage compétitif avec répulsion

Face à ce constat, une solution possible pour améliorer le processus d'apprentissage réside dans l'ajout d'un effet de répulsion de façon à éloigner du neurone gagnant l'ensemble des neurones n'appartenant pas à son voisinage d'adaptation. Cette procédure permet au réseau de s'étendre plus rapidement dans l'espace de représentation, réduisant du même coup la durée nécessaire à son organisation durant la phase d'ordre global. Elle permet également une réduction du niveau du voisinage d'adaptation utilisé en début d'apprentissage. Le déplacement des neurones repoussés se fait dans la direction opposée au déplacement du neurone gagnant.

Ainsi, dans le cadre de l'approche avec répulsion, la fonction de voisinage d'adaptation $h_{ci}(t)$ est définie comme suit :

$$h_{ci}(t) = \begin{cases} 1; & \text{si } d_v(c,i) \leq N_c(t) \\ -1; & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

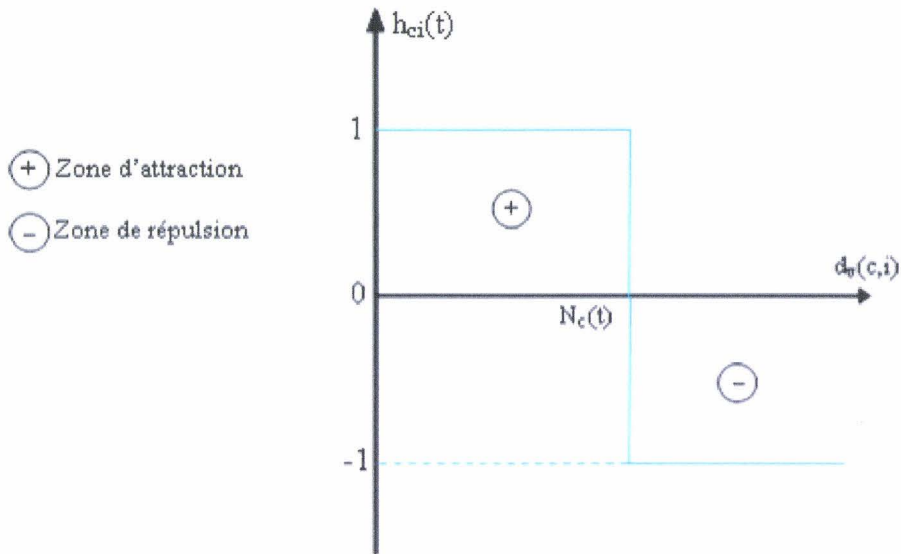


Figure 3.8 Fonction de voisinage d'adaptation pour l'apprentissage compétitif avec répulsion

La fonction de voisinage de l'apprentissage compétitif avec répulsion se compose de 2 zones d'interaction distinctes : une zone d'attraction pour les neurones appartenant au voisinage d'adaptation du neurone gagnant et une zone de répulsion au-delà, pour le restant des neurones du réseau (cf. figure 3.8). La zone d'attraction correspond à des valeurs positives pour le coefficient d'adaptation alors que la zone de répulsion est définie par des valeurs négatives du coefficient d'adaptation, appelé coefficient de répulsion dans cette zone. Le coefficient de répulsion provoque l'éloignement des neurones par rapport au neurone gagnant.

Durant toute la durée de l'apprentissage, l'effet répulsif est beaucoup moins important que l'effet d'attraction afin d'éviter une divergence du réseau, notamment dans les premières itérations de l'apprentissage qui correspondent à la phase d'ordre global.

3.2.3. Algorithme pour l'apprentissage compétitif avec répulsion

- 1- Initialisation des vecteurs-poids m_i pour $t = 0$.
- 2- A l'itération de rang t , un point x_j , qu'on notera $x(t)$, est choisi de manière aléatoire dans l'espace d'entrée. $x(t)$ est présenté à l'entrée du réseau pour être comparé à tous les vecteurs-poids $m_i(t)$ en utilisant la distance euclidienne $d_e(x(t), m_i(t))$. Le neurone pour lequel la distance $d_e(x(t), m_i(t))$ est la plus petite est déclaré vainqueur : son indice est noté c et son vecteur-poids $m_c(t)$, avec $c = \underset{i}{\operatorname{Argmin}} \{ d_e(x(t), m_i(t)) \}$.

3- Le vecteur-poids du neurone vainqueur ainsi que ceux des neurones appartenant à son voisinage d'adaptation sont adaptés de manière à se rapprocher du point présenté en entrée. A l'inverse, les vecteurs-poids des neurones n'appartenant pas au voisinage du neurone vainqueur sont éloignés par rapport au point présenté en entrée. Ainsi, si $m_c(t)$ est le vecteur-poids du neurone vainqueur d'indice c pour l'entrée $x(t)$, la règle d'apprentissage dans son voisinage est définie comme suit :

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) \leq N_c(t) \\ m_i(t) + (\alpha(t)/\theta)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

θ est un coefficient de pondération qui réduit le coefficient d'adaptation dans la zone répulsive ($\theta \gg 1$) car une trop forte répulsion cause des divergences dans l'adaptation du réseau.

L'apprentissage compétitif avec répulsion est semblable à l'apprentissage compétitif simple par le fait qu'on peut également y distinguer deux phases d'adaptation : une phase d'ordre global qui évolue de façon progressive au fur et à mesure que l'apprentissage s'achemine vers une phase d'ordre local. Le coefficient de répulsion est très petit face au coefficient d'attraction et décroît également au cours des itérations successives, ce qui implique que l'effet de répulsion n'est véritablement visible qu'en début de l'apprentissage, durant la phase d'ordre global. Son effet deviendra imperceptible au cours de la phase d'ordre local où son action n'est plus primordiale puisque les neurones se sont déjà spécialisés par rapport à des zones distinctes du nuage de points.

L'effet de répulsion accélère la phase d'ordre global. La répulsion aide les neurones du réseau 3D-SOM à se spécialiser beaucoup plus rapidement par rapport à des zones particulières du nuage de points en rapprochant les neurones des zones qu'ils vont représenter dès le début de l'apprentissage.

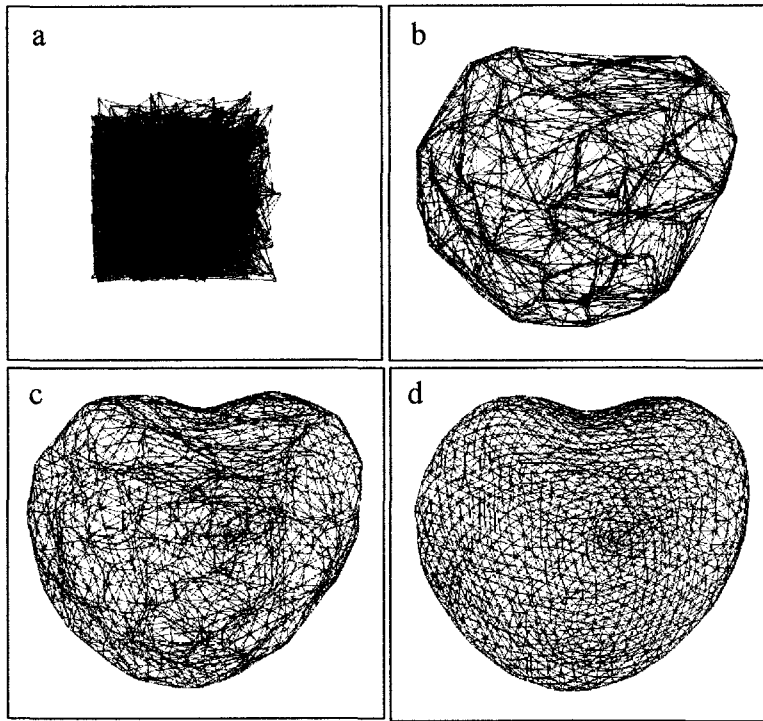


Figure 3.9 Processus d'adaptation d'un réseau 3D-SOM composé de 5122 neurones. La base d'apprentissage est composée de 6402 points représentant la surface d'un cœur.

$$\alpha(0)=0.5, D=0.3, N_c(0)=1000, d=0.6, \theta=5000.$$

a- initialisation aléatoire de tous les vecteurs-poids,

b- état du réseau 3D-SOM après 1000 itérations du processus,

c- état du réseau 3D-SOM après 10000 itérations du processus,

d- adaptation finale après 18000 itérations du processus.

La figure 3.9 illustre le processus d'adaptation d'un réseau 3D-SOM pour un apprentissage compétitif avec répulsion. On y remarque, de façon explicite, l'intérêt de l'effet de répulsion sur le temps total nécessaire à l'adaptation complète du réseau 3D-SOM par comparaison avec les résultats obtenus avec l'apprentissage compétitif simple (20000 itérations, cf. figure 3.7). Cette différence est due au fait qu'on a réduit les coefficients de l'algorithme $\alpha(t)$ et $N_c(t)$ et que les neurones se répartissent et s'organisent plus rapidement grâce à l'effet répulsif. Les neurones se rapprochent des zones qu'ils représentent plus rapidement.

4. Initialisation du réseau 3D-SOM

4.1. Initialisation aléatoire

L'initialisation des vecteurs-poids peut se faire selon deux stratégies. Une première consiste en une initialisation aléatoire (cf. figure 3.10a), de telle sorte que l'apprentissage débute à partir d'une structure totalement désorganisée. Dans ce cas de figure, il est préférable que la majorité des neurones participent à l'apprentissage durant les premières itérations du processus d'adaptation.

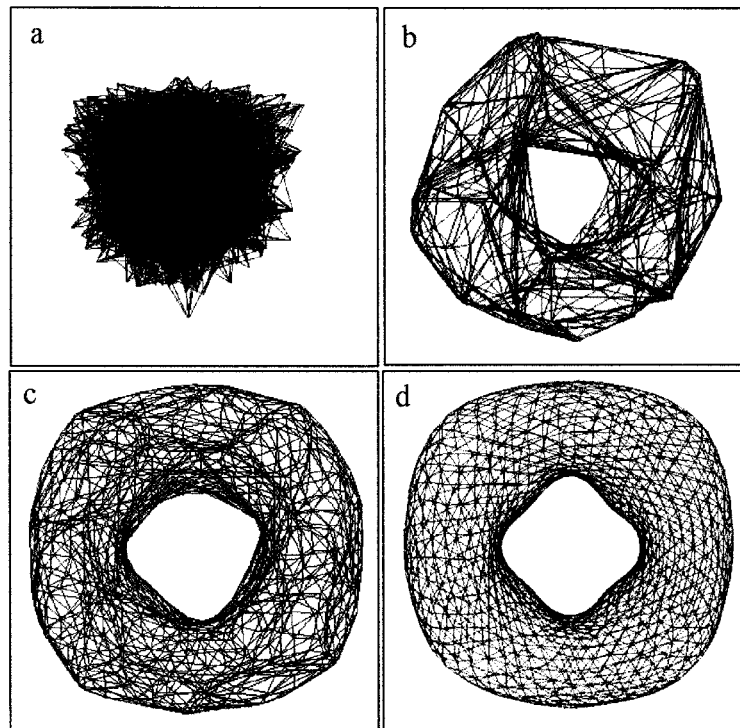


Figure 3.10 Processus d'adaptation d'un réseau 3D-SOM composé de 1154 neurones, pour un nuage de 7381 points représentant une pièce torique.

$$\alpha(0)=0.4, D=0.3, N_c(0)=700, d=0.6, \theta=5000.$$

a- initialisation aléatoire des vecteurs-poids,

b- état du réseau 3D-SOM après 2000 itérations du processus,

c- état du réseau 3D-SOM après 7000 itérations du processus,

d- adaptation finale après 15000 itérations du processus.

Avec un voisinage d'adaptation et un coefficient d'apprentissage importants, la phase d'ordre global dure relativement longtemps, ce qui permet au réseau de se structurer de manière

cohérente par rapport aux entrées malgré la totale désorganisation initiale du réseau. Le coefficient d'apprentissage et le voisinage d'adaptation, importants au début du processus d'adaptation, vont ensuite décroître de manière graduelle afin d'assurer un déploiement correct du réseau.

4.1. Initialisation « pré-établie »

Une autre stratégie possible consiste à débiter l'apprentissage avec une forme d'architecture déjà établie telle un icosaèdre, un cube ou toute autre structure géométrique de départ obtenue après différentes étapes de subdivision totale.

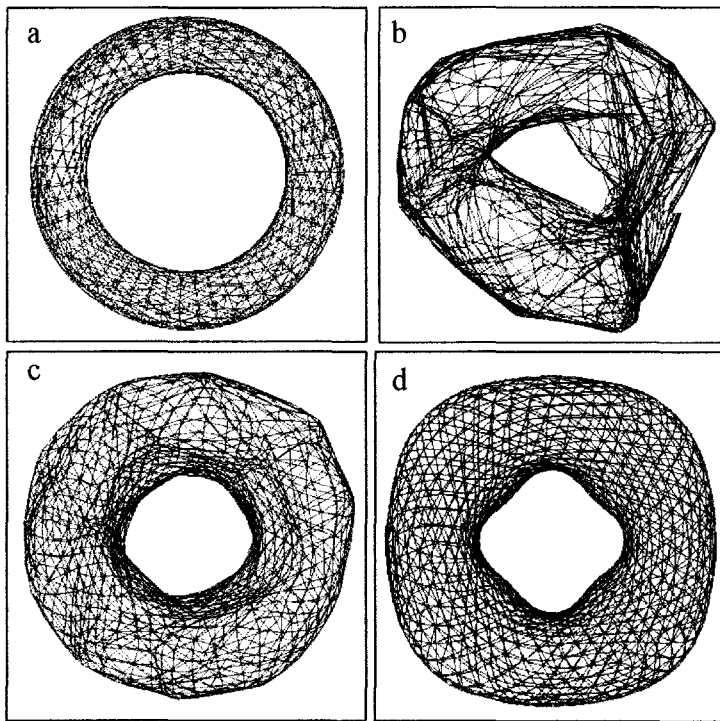


Figure 3.11 Processus d'adaptation d'un réseau 3D-SOM composé de 1154 neurones, sur un nuage de 7381 points représentant une pièce torique.

$$\alpha(0)=0.2, D=0.3, N_c(0)=300, d=0.6, \theta=5000.$$

a- initialisation du réseau suivant la forme de sa structure de départ,

b- état du réseau 3D-SOM après 1000 itérations du processus,

c- état du réseau 3D-SOM après 6000 itérations du processus,

d- d'adaptation finale après 10000 itérations du processus.

A l'instant d'initialisation, les vecteurs-poids des neurones prennent les valeurs des coordonnées des sommets des mailles correspondant à la forme initiale. Il n'y a donc pas d'initialisation aléatoire de ces derniers. Cette configuration de base a pour effet de réduire considérablement les déplacements nécessaires aux neurones pour représenter de manière globale les points présentés en entrée. Le coefficient d'apprentissage et le voisinage d'apprentissage n'ont pas besoin de prendre des valeurs importantes pour garantir la représentation correcte du nuage de points.

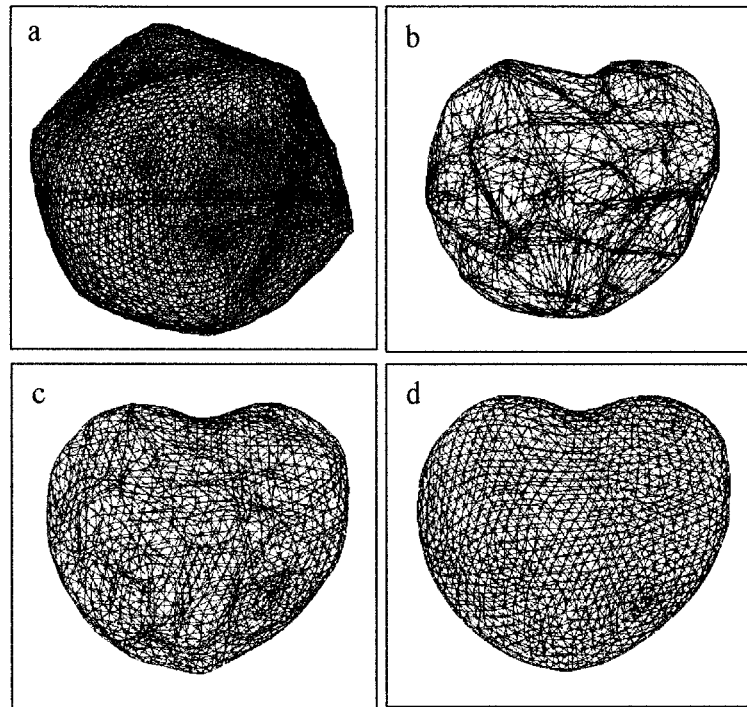


Figure 3.12 Processus d'adaptation d'un réseau 3D-SOM composé de 5122 neurones. En entrée nous avons un nuage composé de 6402 points représentant la surface d'un cœur.

$$\alpha(0)=0.2, D=0.3, N_c(0)=500, d=0.6, \theta=5000.$$

a- initialisation du réseau avec sa structure de départ,

b- état du réseau 3D-SOM après 1000 itérations du processus,

c- état du réseau 3D-SOM après 7000 itérations du processus,

d- adaptation finale après 12000 itérations du processus.

Le fait de partir d'une structure pré-établie nous permet de réduire le voisinage d'adaptation au début de l'apprentissage, du fait que le réseau a déjà une structure cohérente vis-à-vis de la répartition des points d'entrée dans l'espace de représentation. Le temps nécessaire

pour que l'ordre global se mette en place de façon cohérente est, de fait, inférieur au temps nécessaire pour une procédure avec initialisation aléatoire.

Sur les figures 3.10 et 3.11, on constate que le réseau 3D-SOM converge plus rapidement vers la forme finale dans le cas où l'on débute l'apprentissage avec une forme d'architecture pré-établie en comparaison avec le cas d'une initialisation aléatoire des vecteurs-poids. Il faut 15000 itérations d'adaptation pour atteindre la représentation finale dans le cas d'une initialisation aléatoire, par contre il n'en faut que 10000 dans le cas d'une initialisation à partir d'une structure pré-établie. Le choix des paramètres d'adaptation est important car s'ils sont mal choisis, par exemple s'ils sont trop grands, l'avantage de commencer l'adaptation avec une structure préformée est perdu et l'adaptation revient à initialiser le réseau 3D-SOM de manière aléatoire.

La figure 3.12 illustre également, à l'aide d'un second exemple, le gain de temps d'apprentissage obtenu lorsqu'on part d'une structure pré-établie pour atteindre la forme finale de l'objet à l'aide du réseau (12000 itérations) par comparaison avec le cas d'une initialisation aléatoire (18000 itération, illustré en figure 3.9). L'organisation initiale du réseau accélère la procédure d'ordre global.

5. Etude de l'effet des paramètres d'adaptation sur les 3D-SOM

Les 3D-SOM obéissent aux mêmes contraintes que les cartes auto-organisatrices classiques en terme de sensibilité par rapport aux valeurs des paramètres de l'algorithme $N_c(t)$ et $\alpha(t)$. La figure 3.13 présente plusieurs exemples qui permettent de juger l'importance de l'ajustement de ces paramètres pour permettre une bonne reconstruction d'une forme d'objet 3D à partir de son nuage de points. Le réseau utilisé pour cet exemple est composé de 5122 neurones et le nuage de points en entrée, composé de 62171 points, représente la surface d'un vase. L'apprentissage a commencé à partir d'un réseau initialisé de manière aléatoire avec un algorithme de type compétitif simple, donc sans effet de répulsion.

5.1. Sensibilité de l'apprentissage par rapport au coefficient d'apprentissage $\alpha(t)$

Si le coefficient d'apprentissage $\alpha(t)$ diminue trop vite par rapport au niveau du voisinage d'apprentissage, on aboutit à une sorte d'adaptation incomplète, c'est-à-dire que les neurones du réseau commencent à se structurer de manière cohérente au début de l'apprentissage mais au fur et à mesure de la diminution du coefficient d'apprentissage $\alpha(t)$ les neurones n'ont plus la

« force » nécessaire pour atteindre une position suffisamment proche des points présentés en entrée. Ceci se remarque sur les figures 3.13b et 3.13c, où on constate qu’après un début d’apprentissage cohérent du réseau, celui-ci finit par se figer, c'est-à-dire que les neurones ont tendance à s’immobiliser, dans des positions qui ne seraient qu’intermédiaires au cours d’un apprentissage normal (cf. figures 3.13a).

5.2. Sensibilité de l’apprentissage par rapport au niveau du voisinage d’adaptation $N_c(t)$

De la même façon que pour les cartes 2D, si le niveau du voisinage d’adaptation du réseau 3D-SOM diminue trop rapidement ou si le niveau du voisinage d’adaptation à l’itération initiale de rang $t = 0$ est trop faible, le réseau n’a pas la possibilité de se structurer de manière efficace. La phase d’ordre global ne se réalise pas de façon satisfaisante et la couverture de l’espace d’entrée n’est plus garantie, ce qui provoque des distorsions dans le réseau (cf. figures 3.13d et 3.13e).

Au début de l’apprentissage, le niveau de voisinage doit donc être suffisant afin de permettre au réseau de se déployer de manière cohérente. La taille du voisinage d’adaptation de départ doit être proportionnelle au nombre de neurones présents dans le réseau 3D-SOM.

Il est par ailleurs impératif que la phase d’ordre local soit suffisamment longue pour permettre au réseau d’appréhender correctement les détails de la surface de l’objet à reconstruire.

De ce fait, un bon choix des paramètres d’adaptation est primordial pour permettre une bonne reconstruction. Cependant il n’existe pas de règle formelle permettant de déterminer les valeurs optimales des paramètres d’adaptation. Dans notre cas, nous avons procédé suivant une approche empirique. Les valeurs des paramètres de départ ($\alpha(0)$, D , $N_c(0)$, d) sont données sur la figure 3.13.

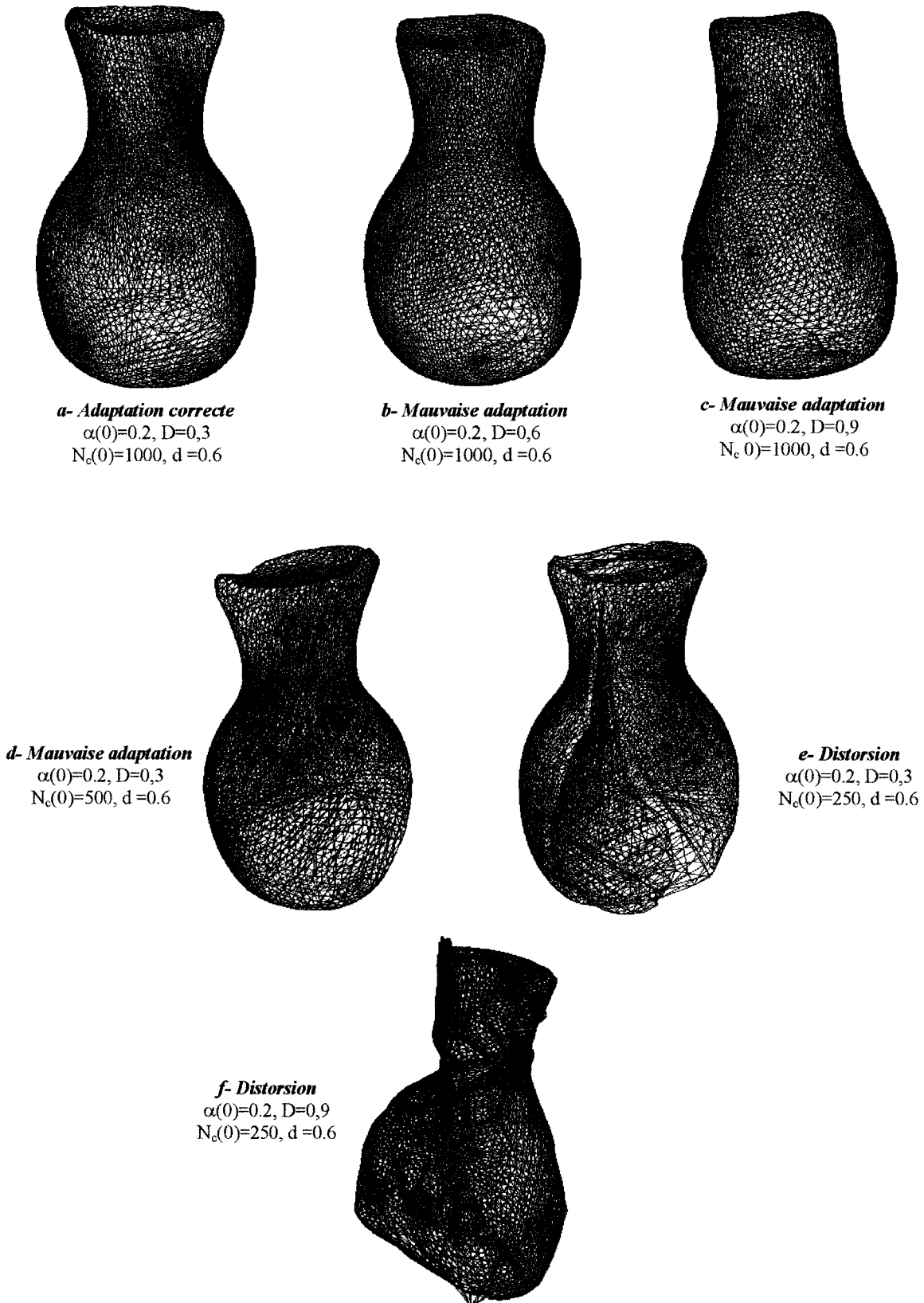


Figure 3.13 Exemples de reconstruction montrant l'influence des paramètres d'adaptation sur les 3D-SOM

6. Chaîne d'acquisition/ reconstruction par 3D-SOM

Dans cette partie du chapitre, nous illustrons la procédure complète de reconstruction d'un objet 3D à l'aide des 3D-SOM en partant de la phase d'acquisition du nuage de points à la surface de l'objet jusqu'à sa reconstruction et sa représentation finale. Cette expérience a été réalisée dans le cadre d'une collaboration avec l'Equipe des Systèmes de Vision par Ordinateur de l'Institut d'Informatique Théorique et Appliquée de l'Académie Polonaise des Sciences IITiS PAN à Gliwice. L'acquisition du nuage de points s'est faite en utilisant un scanner 3D de marque Konica-Minolta VI-9i (cf. figure 3.14a).

6.1. Matériel utilisé pour l'acquisition de points

Le VI9i est basé sur le principe de la triangulation laser (cf. figure 3.14b). Une cible est balayée avec des raies de faisceaux laser. Une caméra CCD reçoit la lumière réfléchie par la surface de l'objet. Les mesures 3D de la forme sont ensuite obtenues par calculs (cf. figure 3.14b). Le VI9i mesure 640 x 480 points par balayage. Nous n'aborderons pas ici le principe de ces calculs, nous invitons pour cela le lecteur à consulter les références techniques données par le constructeur (référence : <http://www.konicaminolta-3d.com/products/vi9i/>).

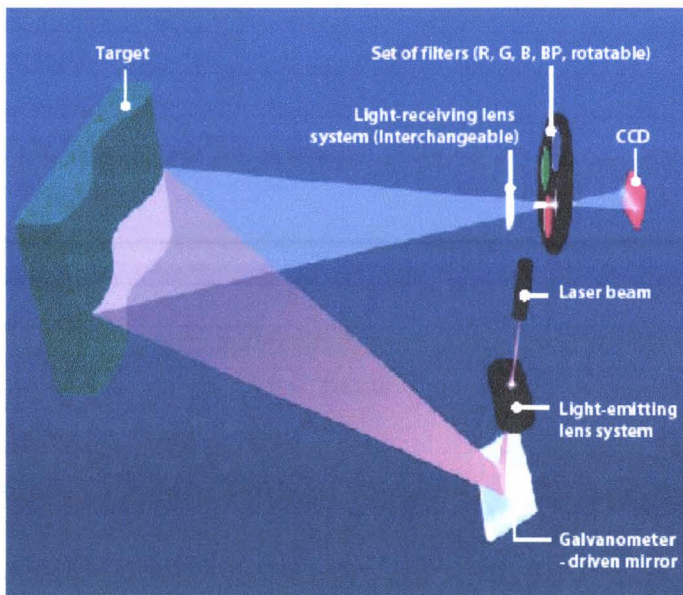
Le VI-9i peut numériser différents volumes (entre 110 x 80 x 40 mm et 1200 x 900 x 750 mm), simplement en sélectionnant l'objectif approprié. Les trois objectifs (telephoto, normal et grand angle) sont livrés en équipement standard et s'intervertissent très facilement.

Ne pesant que 11kg et mesurant 213x413x 271mm, le VI-9i est très compact. La carte mémoire compact-flash permet de sauvegarder plusieurs scans sans être relié à un ordinateur.

Contrairement à d'autres systèmes de numérisation, le VI-9i ne nécessite aucune mise en oeuvre fastidieuse, ni procédure de calibration ou autre type de préparation avant utilisation. Il n'est pas nécessaire d'avancer ou de reculer le scanner pour se positionner à la bonne distance, l'autofocus s'occupe de faire la mise au point automatiquement.



a



b

*Figure 3.14 Principe d'acquisition de points 3D
à l'aide du scanner Konica-Minolta VI-9i*

Pour illustrer la chaîne complète d'acquisition et de reconstruction par un exemple concret, nous avons utilisé la maquette d'un véhicule routier (maquette au 1/18, cf. figure 3.15a). La surface de cette maquette a été échantillonnée à l'aide du scanner VI-9i (cf. figures 3.15b), et a généré un nuage de 11121 points représentant la surface de ce véhicule (cf. figures 3.15c).

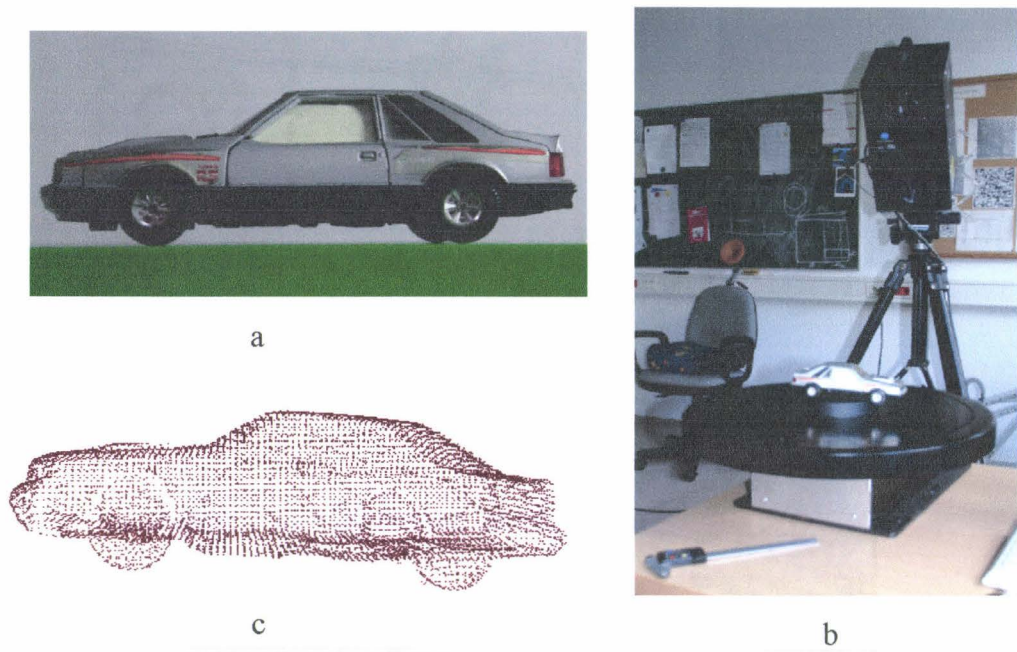
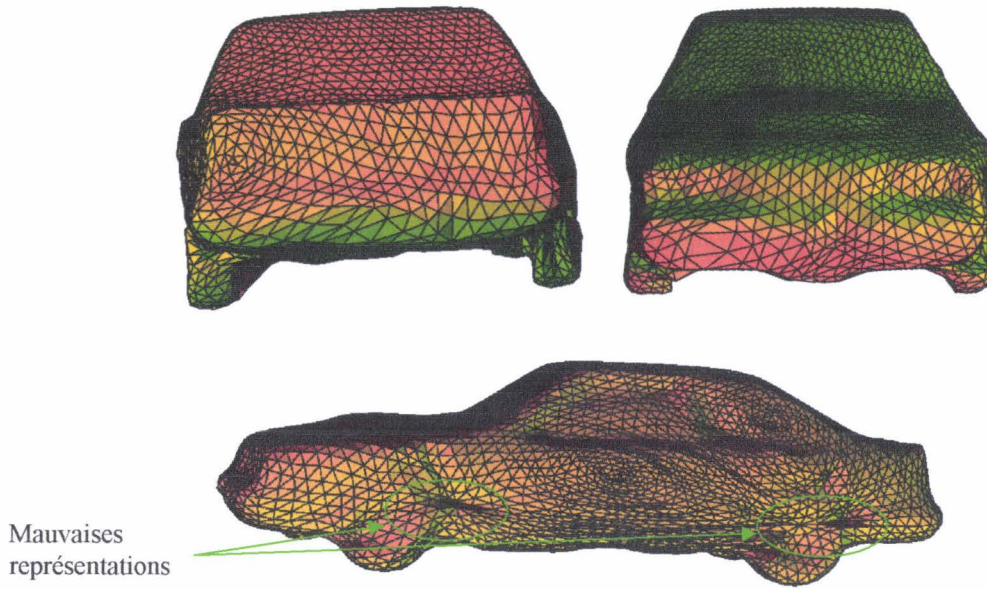


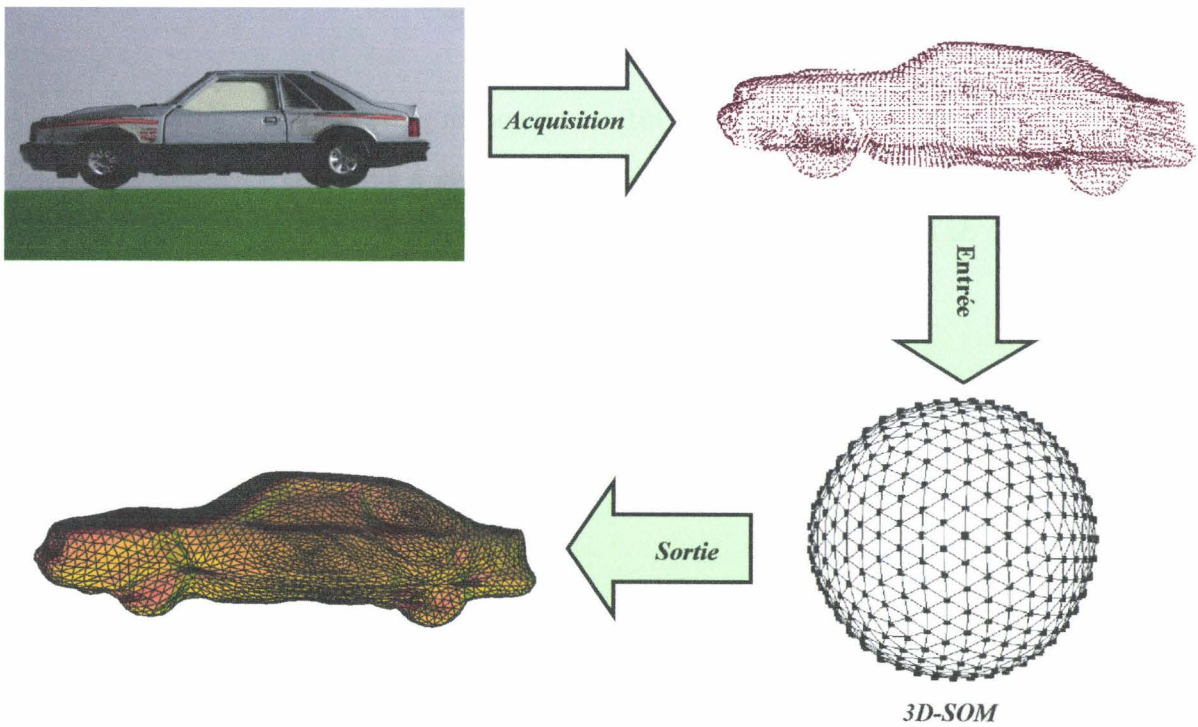
Figure 3.15 Processus d'acquisition, a– maquette du véhicule, b– phase d'acquisition à l'aide du matériel VI-9i, c – le nuage de points 3D acquis à la surface du véhicule.

6.2. Mise en œuvre du réseau 3D-SOM et résultats

Pour la modélisation et représentation de ce véhicule, nous utilisons un réseau 3D-SOM composé de 5122 neurones. La figure 3.16a illustre le résultat de la reconstruction du véhicule, avec différentes vues du maillage obtenu.



a- Différentes vues du véhicule reconstruit



b- Illustration de la chaîne d'acquisition/ reconstruction par 3D-SOM

Figure 3.16 Etapes du processus de reconstruction du véhicule

La figure 3.16b illustre les principales étapes du processus de reconstruction à l'aide du réseau 3D-SOM. Le résultat final représente le maillage du véhicule, nous constatons que la carte a pu reproduire la forme générale de l'objet, sauf au niveau de la jonction entre la carrosserie et les roues du véhicule où on remarque des zones de surface moins bien représentées, ceci est dû au fait qu'il existe des zones où il n'y a pas de points entre la carrosserie et les roues ce qui provoque une anomalie de reconstruction. Ce problème est inhérent au matériel d'acquisition.

7. Conclusion

Dans ce chapitre, nous avons introduit les concepts essentiels de l'architecture, de l'algorithme et les stratégies d'adaptation des 3D-SOM. Différentes simulations ont permis de démontrer l'efficacité du modèle présenté.

Le fait de partir d'une structure complètement désorganisée pour aboutir à une représentation exacte de la forme à modéliser nous renseigne sur la capacité d'adaptation des réseaux 3D-SOM.

Le modèle de base du réseau 3D-SOM est cependant perfectible et nous proposons dans le chapitre suivant différentes évolutions possibles destinées à améliorer l'adéquation des 3D-SOM au problème de la reconstruction et de la représentation de formes d'objets 3D à partir de nuages de points.

Améliorations du processus d'adaptation des 3D-SOM

1. Introduction

Dans le chapitre 3, nous avons introduit les 3D-SOM en termes d'architecture, d'algorithmes d'apprentissage et de stratégies d'adaptation. Différents exemples de reconstruction de surfaces 3D à partir de nuages de points ont permis d'évaluer la qualité de la représentation des formes d'objets tridimensionnels obtenues avec les 3D-SOM. Néanmoins, certaines améliorations peuvent être apportées au niveau des algorithmes d'adaptation pour augmenter encore l'efficacité de reconstruction des 3D-SOM.

Les algorithmes d'apprentissage présentés dans le chapitre 3 s'inspirent des lois d'adaptation déjà existantes dans le domaine des cartes auto-organisatrices. Ces algorithmes ont été adaptés pour permettre une bonne adéquation avec l'architecture des 3D-SOM. Cependant, malgré l'efficacité globale reconnue pour ces lois d'adaptation, les cartes auto-organisatrices génèrent parfois des défauts de représentation et les 3D-SOM n'échappent pas à ce constat. Les anomalies constatées ne sont généralement pas très importantes et ne constituent pas des défauts graves dans le cas des applications de visualisation ou de classification de données, domaines dans lesquels les cartes auto-organisatrices sont largement utilisées et où l'aspect visuel de la représentation de la distribution des données n'est pas primordial.

A l'inverse, dans le domaine de la reconstruction et de la représentation de formes d'objets, l'aspect visuel du résultat reflète directement la qualité de la reconstruction qui doit être aussi représentative que possible de la surface originale de l'objet. Un défaut ou une anomalie d'apprentissage empêche la représentation correcte du nuage de points présenté en entrée du processus de reconstruction. Il est donc nécessaire d'apporter des améliorations aux cartes 3D-SOM pour résorber les éventuels défauts d'adaptation qui pourraient apparaître lors du processus de reconstruction.

Ce chapitre a pour but de présenter différentes solutions et améliorations possibles pour le processus de reconstruction à l'aide des 3D-SOM. Les améliorations apportées se situent au niveau des algorithmes d'apprentissage et de l'architecture topologique de la carte.

2. Prise en compte de la densité de points dans le processus d'adaptation

Les améliorations apportées au réseau 3D-SOM qui sont présentées dans ce paragraphe sont inspirées des travaux de Fritzke [Fri 97] sur les cartes auto-organisatrices et l'apprentissage compétitif. C'est ainsi que nous préconisons l'utilisation d'une méthode qui prend en compte les densités locales de points dans le nuage présenté en entrée du réseau dans le but de permettre une meilleure adaptation de réseau et ainsi une meilleure représentation des surfaces reconstruites.

Cette approche est guidée par le fait, qu'en général, le nombre de points acquis à la surface de l'objet au cours de son échantillonnage est inversement proportionnel à la régularité de la surface. De façon schématique, cela signifie que peu de points sont acquis sur des surfaces relativement planes mais le nombre d'acquisitions doit être sensiblement augmenté lorsque la surface à échantillonner présente localement des formes complexes ou des concavités prononcées (cf. figure 4.1). Le fait d'augmenter localement le nombre de points échantillons lors du processus de numérisation de l'objet 3D, pour les parties présentant des formes complexes permettra de restituer plus fidèlement et plus facilement la forme complète de l'objet lors du processus de reconstruction.

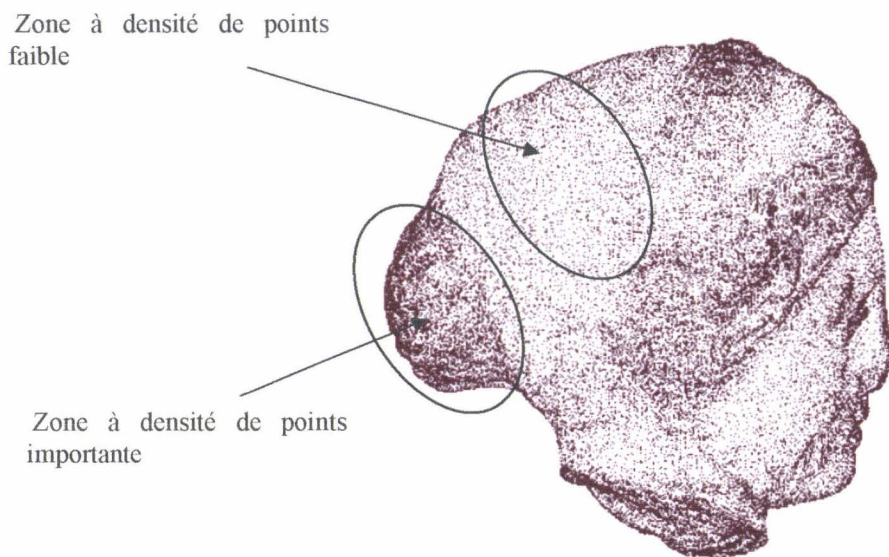


Figure 4.1 Tête de statue sous la forme d'un nuage de points révélant différentes densités de points suivant la complexité locale de la surface

Dans ce contexte, même si par nature les 3D-SOM permettent de façon intrinsèque la prise en compte de la distribution des points à l'intérieur du nuage présenté en entrée du réseau pour réaliser un processus d'auto-adaptation, un renforcement de la loi d'adaptation permettra d'accentuer la prise en compte de la caractéristique de densité locale des points au profit des zones denses pour en permettre une meilleure représentation.

Afin de prendre en compte la caractéristique de densité locale des points dans le processus d'adaptation, nous proposons de moduler l'adaptation des neurones à l'aide d'un paramètre supplémentaire représentant une quantification de la densité locale dans le nuage de points, estimation qui s'ajustera au fur et à mesure de l'apprentissage et dont une valeur est établie pour chaque neurone.

L'estimation de la densité locale des points en entrée se fait par l'intermédiaire d'un compteur qui comptabilise le nombre de fois qu'un neurone est élu vainqueur pour un point présenté en entrée du réseau. Ce comptage est fait systématiquement pour chaque nouveau point $x(t)$ présenté et l'unité d'incrémentation s'ajoute aux valeurs précédentes déjà accumulées pour le même neurone gagnant d'indice c . La valeur atteinte par le compteur propre à un neurone d'indice i est mémorisée dans un compteur neuronal noté μ_i qui est mis à jour à chaque pas d'apprentissage (itération de rang t) de la façon suivante :

$$\mu_i(t+1) = \begin{cases} \mu_i(t) + 1; & \text{si } i = c \\ \mu_i(t); & \text{si } i \neq c \end{cases}$$

i spécifie l'indice d'un neurone quelconque du réseau et c spécifie l'indice du neurone gagnant, avec $i=1,2,\dots,M$.

Les compteurs neuronaux μ_i qui s'incrémentent tout au long de l'apprentissage pour tous les neurones élus neurones gagnants permettent de savoir quel est le neurone qui a été le plus souvent élu vainqueur. Le fait qu'un neurone soit élu vainqueur plus souvent que les autres signifie qu'il est représentatif d'une région à très forte densité de points car la probabilité qu'un neurone soit élu vainqueur est fortement liée à la densité de points de la région qu'il représente. En effet, plus il y a de points proches d'un neurone, plus ces chances de gagner la compétition neuronale sont nombreuses, ce qui entraîne une incrémentation plus rapide de son compteur par rapport à ceux des autres neurones du réseau. De façon concrète, plus le niveau atteint par le compteur neuronal d'un neurone est grand, plus la densité des points de la zone qu'il représente par rapport au nuage est importante.

Par ce moyen, nous pouvons quantifier la densité de points dans chaque région du nuage de points de la surface à reconstruire. L'ajout de ce compteur neuronal aux paramètres

d'apprentissage permettra une amélioration de l'adaptation locale de chaque neurone en liaison avec la partie du nuage de point qu'il représente.

L'algorithme avec prise en compte de la notion de densité de points utilise la loi d'adaptation suivante :

$$m_i(t+1) = \begin{cases} m_i(t) + h_{ci}(t)(\mu_i(t)/\lambda + \alpha(t))(x(t) - m_i(t)); & \text{si } d_v(c,i) \leq N_v(t) \\ m_i(t) + h_{ci}(t)(\alpha(t)/\theta)(x(t) - m_i(t)); & \text{si } d_v(c,i) > N_v(t) \end{cases}$$

La valeur du coefficient de pondération λ est déterminée de manière empirique par l'utilisateur. Ce paramètre λ permet de limiter l'effet du compteur neuronal, de façon à ne pas fausser le processus d'adaptation pour de grandes valeurs du compteur. Pour l'exemple de la figure 4.2, la valeur de λ est fixée à 5000.

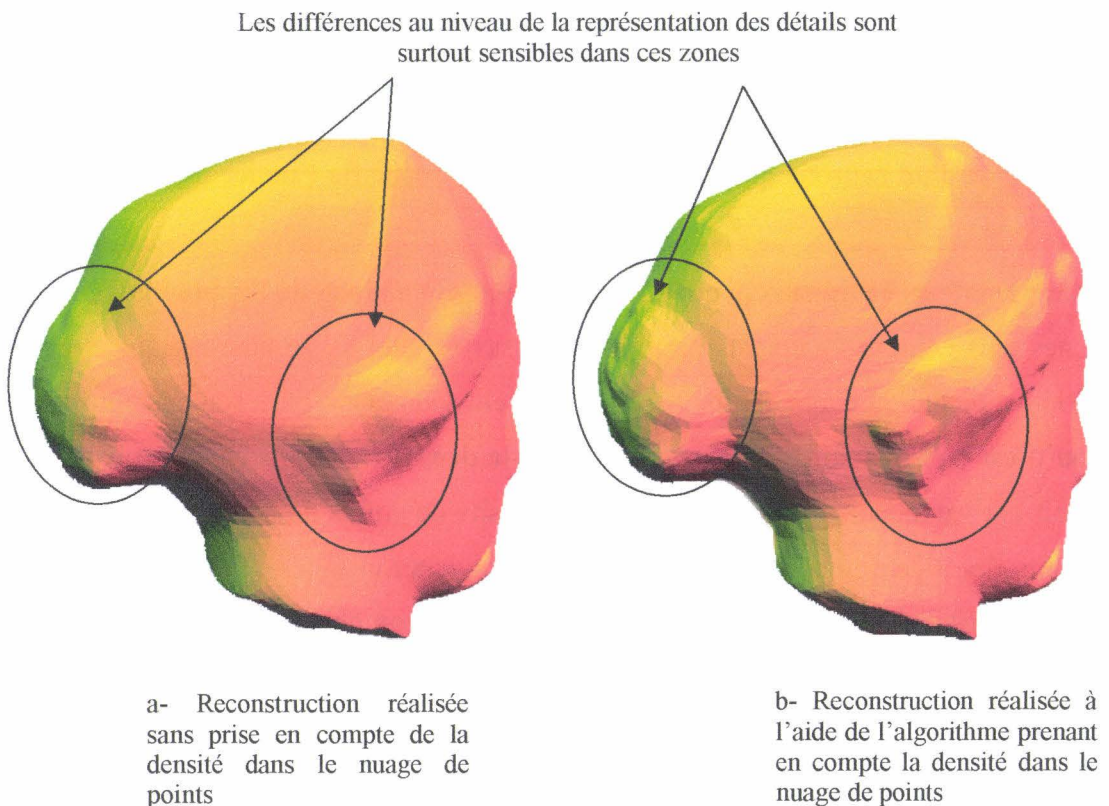


Figure 4.2 Exemple de reconstruction avec prise en compte de la notion de densité de points

Sur la figure 4.2b, nous présentons un exemple d'adaptation réalisé à l'aide de l'algorithme avec prise en compte de la densité des points en entrée. L'adaptation des neurones s'effectue de manière différenciée en fonction des différentes densités de points dans les différentes zones de la surface. Par comparaison avec la reconstruction utilisant un algorithme classique sans prise en compte de la densité locale (cf. figure 4.2a), nous remarquons une meilleure représentation des détails surfaciques. On remarque en particulier que l'algorithme est plus efficace dans les zones révélant de nombreux détails. Les zones où la surface ne présente pas beaucoup de relief sont par contre plus ou moins équivalentes au niveau de la représentation.

3. Défauts d'adaptation

A l'issue du processus d'adaptation, il arrive parfois que les cartes auto-organisatrices viennent à représenter des zones vides de l'espace d'entrée, c'est à dire ne contenant aucune donnée de la base d'apprentissage. Ce comportement n'est pas forcément pénalisant si la carte est utilisée dans un but de classification ou de modélisation des données présentées en entrée, car dans ce cas, l'utilisateur analyse essentiellement la disposition finale des neurones dans l'espace de représentation et non pas la visualisation des liens de voisinage entre neurones, ni la forme du maillage pour les zones non représentatives (« vides »). Ce défaut apparent d'adaptation devient cependant rédhibitoire pour une application liée à la reconstruction de surfaces telle que peut le montrer l'exemple sur la figure 4.3.

Sur cet exemple, nous constatons effectivement que certains neurones de la carte 2D viennent à représenter des parties qui n'existent pas au niveau de la surface apparente de la surface présentée en entrée du réseau sous forme d'un ensemble de points 2D. Plus précisément, les coins où la forme présente de fortes discontinuités, ne sont pas fidèlement représentés par l'architecture de la carte SOM.

D'une manière générale, les cartes SOM éprouvent des difficultés à reproduire fidèlement la forme d'une surface lorsqu'elle présente localement des zones à forte concavité, comme par exemple les coins pour la forme de la figure 4.3.

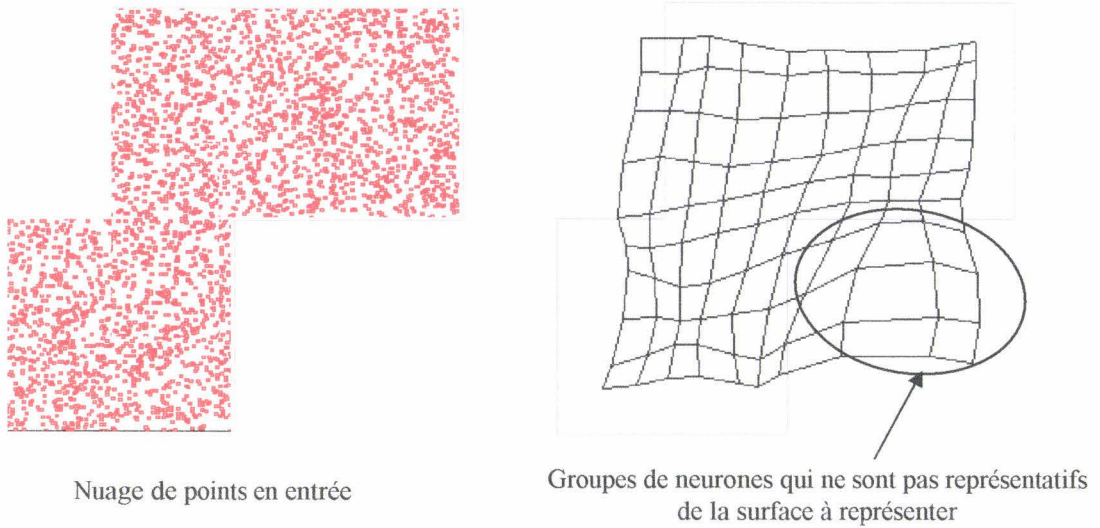


Figure 4.3 : Représentation de la surface d'un objet plan à l'aide d'une carte de Kohonen 2D : défaut apparent d'adaptation

Ce défaut d'adaptation local peut être dû à plusieurs facteurs : mauvais choix des paramètres d'adaptation, voisinage ou coefficient d'apprentissage qui décroît trop rapidement ou qui ne sont pas suffisamment grands au début de l'apprentissage.

Une autre cause possible pour ce type de défaut d'adaptation est liée au fait que les neurones qui, pendant la phase d'ordre local du processus d'apprentissage, sont appelés à être représentatifs d'une zone présentant une surface à forte concavité se retrouvent généralement attirés de manière pratiquement égale de part et d'autre des bords de cette zone. Cela signifie que, dans cette zone, un même neurone est susceptible d'être élu indifféremment vainqueur pour des points d'entrée différents, d'origines bien distinctes, appartenant par exemple à des bords opposés de la zone (cf. figure 4.4). Ce phénomène local d'attraction mutuelle provoque un effet de neutralisation de l'adaptation. Il se manifeste par le fait que les neurones restent éloignés des positions respectives de ces points particuliers présentés en entrée. Ces neurones ne peuvent pas s'adapter de façon complète durant la phase d'ordre local et donc ne peuvent pas véritablement représenter de façon fidèle les zones à forte concavité de la surface à reconstruire à la fin du processus d'apprentissage.

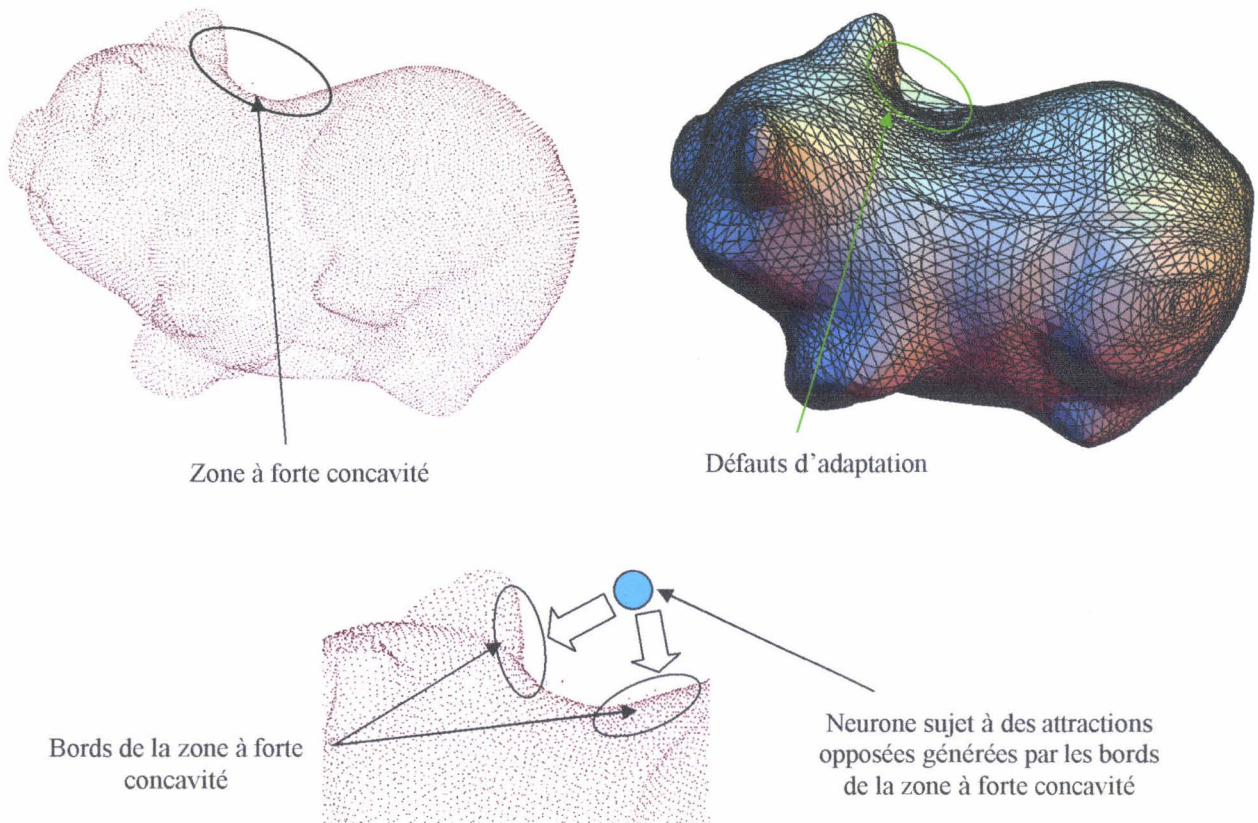


Figure 4.4 Défauts d'adaptation dans un réseau 3D-SOM

Dans la suite de ce chapitre, pour tenter de répondre au problème des défauts d'adaptations pour les surfaces à fortes concavités locales, des solutions sont proposées à partir de différentes approches basées respectivement sur les principes de réadaptation, de subdivision locale et d'adaptation locale. Le but de ces méthodes est de traiter certaines parties de l'architecture de la 3D-SOM comme des cartes auto-organisatrices individuelles et indépendantes. Nous verrons ainsi comment ces différentes approches permettent d'améliorer, de façon non-supervisée, la représentation locale du nuage de points présenté en entrée du réseau dans les zones où apparaissent des défauts d'adaptation.

4. Réadaptation

Une des solutions les plus simples pour tenter de répondre au problème des défauts d'adaptation est, à la fin du processus d'apprentissage, de provoquer la réadaptation de la totalité du maillage neuronal. Le fait de traiter l'ensemble de l'architecture permet naturellement de

traiter également les zones affectées par des défauts apparents d'adaptation. La procédure de réadaptation consiste à appliquer un second processus d'adaptation avec des coefficients adéquats en réduisant le voisinage d'adaptation et le coefficient d'apprentissage. Cette deuxième phase d'adaptation, dite de réadaptation, est très rapide en comparaison avec la première phase, du fait que le maillage est déjà largement préformé par rapport à la forme globale de l'objet à reconstruire et que les coefficients de l'algorithme mis en œuvre dans la procédure de réadaptation sont réduits.

Cette méthode se révèle efficace dans la plupart des cas (cf. figure 4.5). Néanmoins il arrive parfois que la procédure de réadaptation n'apporte aucune amélioration apparente à la qualité de la reconstruction pour certaines zones présentant un défaut d'adaptation. De ce fait, il s'avère nécessaire de rechercher une solution en aval du processus d'adaptation, ciblée cette fois-ci uniquement sur les zones où les défauts d'adaptation persistent.

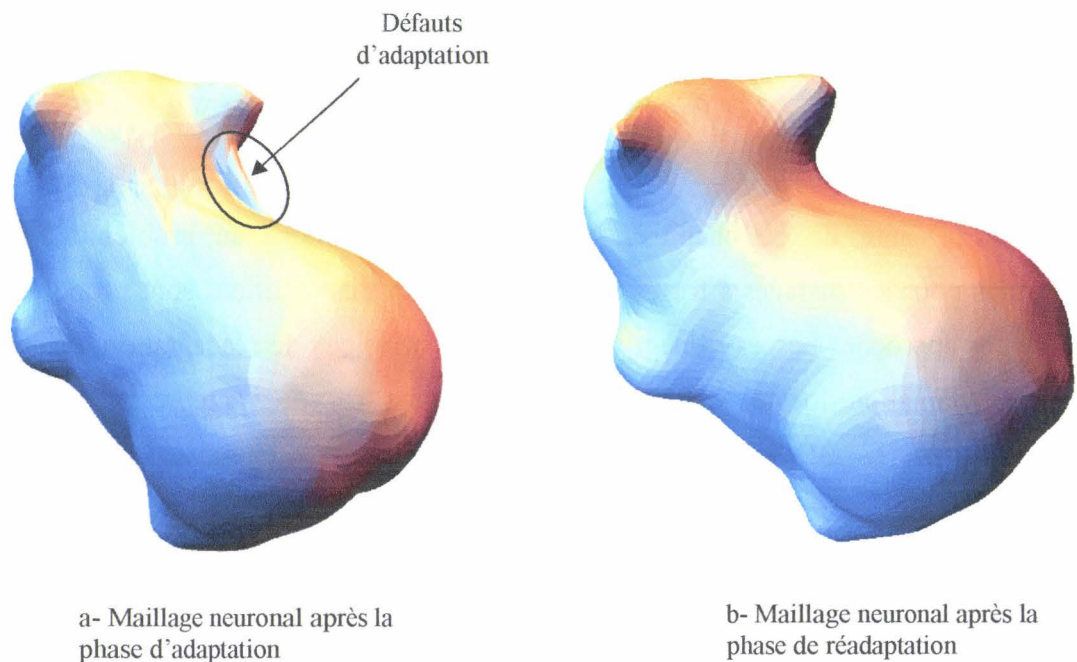


Figure 4.5 Illustration de l'effet de la procédure de réadaptation sur un maillage préformé présentant un défaut d'adaptation locale.

5. Subdivision/Adaptation locale

Il est très facile de constater visuellement que les parties du maillage de l'architecture de la 3D-SOM qui présentent des défauts d'adaptation sont presque systématiquement composées de triangles très allongés par rapport à l'élongation moyenne des autres triangles du maillage (cf. figure 4.6).

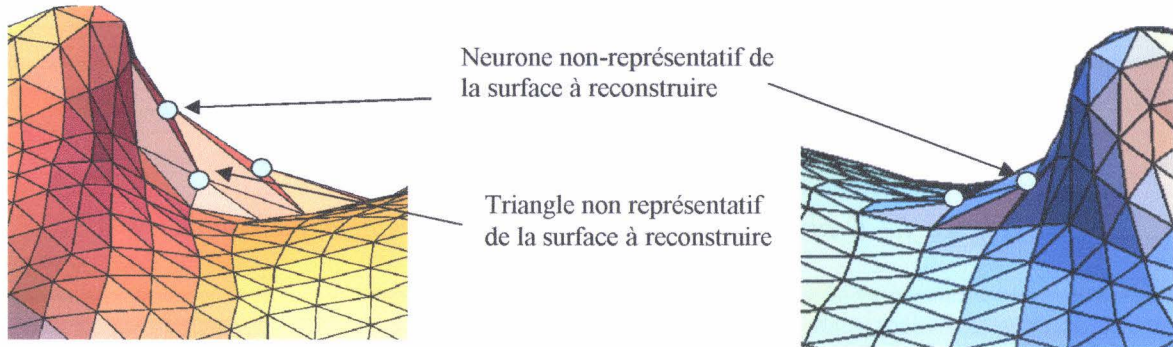


Figure 4.6 Neurones et triangles non-représentatifs

Si des défauts d'adaptation persistent à la fin de l'apprentissage, il est possible de repérer les parties du maillage où des triangles présentent localement des formes plus allongées que la moyenne et donc nécessitent un renforcement de la procédure d'adaptation.

Dans la suite de l'exposé, nous appellerons ces triangles allongés ainsi que les neurones qui les composent, respectivement des triangles et des neurones « non-représentatifs » car ils représentent mal la forme réelle à la surface de l'objet.

5.1. Identification des triangles non-représentatifs

L'identification des triangles non-représentatifs se fait à partir de la mesure de leur élongation. En pratique, ce critère s'est avéré suffisamment discriminant pour distinguer les triangles non-représentatifs des autres triangles du maillage.

Nous définissons de façon simple l'élongation E d'un triangle comme étant le rapport entre la longueur l_A de son côté le plus grand et la longueur l_B de son côté le plus petit : $E = l_A/l_B$ (cf. figure 4.7).

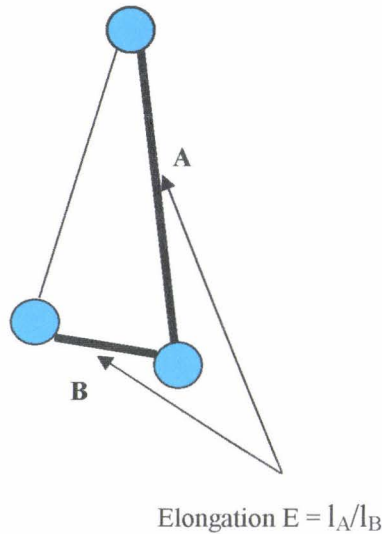


Figure 4.7 Evaluation de l'élongation E d'un triangle quelconque

L'élongation E est calculée pour chaque triangle du maillage à la fin du premier processus d'apprentissage. On calcule alors la valeur moyenne E_m des élongations pour tous les triangles constituant le maillage 3D. On compare ensuite l'élongation de chaque triangle du maillage à la valeur moyenne des élongations : les triangles avec une élongation dépassant de 25% cette moyenne sont classés comme triangles « non-représentatifs ». La valeur du seuil de 25% a été choisie empiriquement après plusieurs essais sur différentes formes d'objets 3D.

Après plusieurs tests réalisés sur différents maillages obtenus après apprentissage de formes de surfaces pour des objets 3D divers, cette méthode s'est avérée très efficace pour repérer les zones où des défauts d'adaptation étaient effectivement présents.

5.2. Subdivision locale

Grâce au critère d'élongation, après avoir identifié à l'intérieur du réseau 3D-SOM les zones du maillage ayant subi une mauvaise adaptation, nous choisissons d'y favoriser le processus d'adaptation en y renforçant l'architecture du réseau ainsi qu'en y renforçant les paramètres d'adaptation.

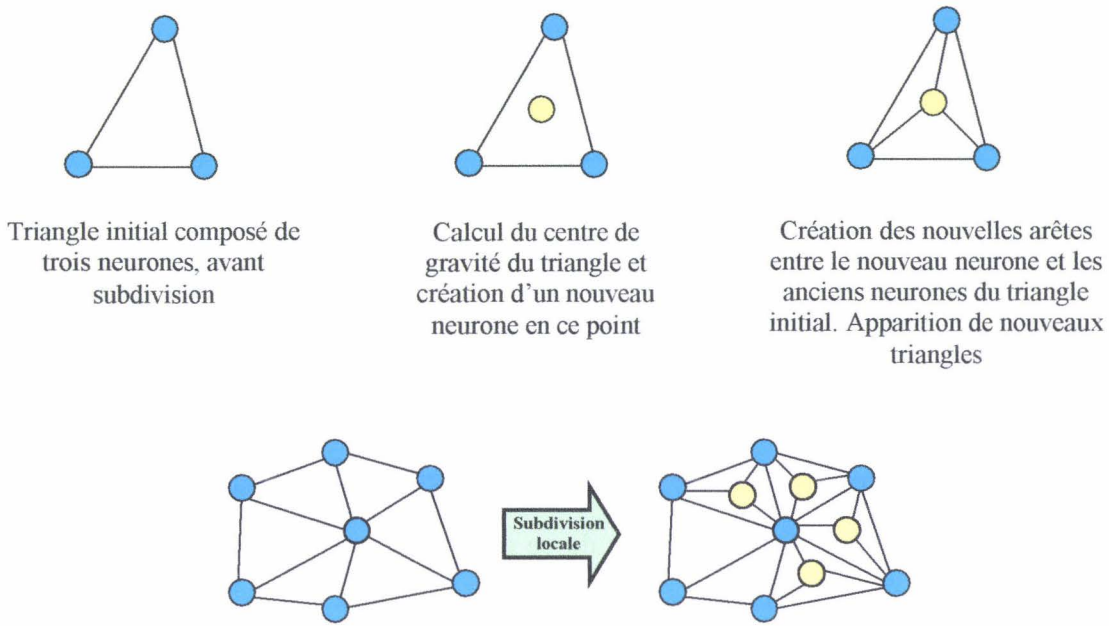


Figure 4.8 Illustration du principe de subdivision locale

Le renforcement de l'architecture s'effectue en ajoutant de nouveaux neurones à l'intérieur des triangles non-représentatifs, ainsi que de nouvelles liaisons de voisinage. Cette transformation s'effectue en suivant un mécanisme de subdivision de l'architecture appelé « subdivision locale ». En effet, l'apparition d'un plus grand nombre de neurones dans les parties où se situent les défauts d'adaptation du maillage, avec des coefficients d'apprentissage localement réactualisés, pourra faciliter et améliorer l'adaptation locale du réseau dans ces zones. Pour illustrer le bien-fondé de cette approche, nous pouvons comparer ce processus à l'échantillonnage d'une courbe : plus il y a de points échantillons, plus la forme réelle de la courbe peut être correctement approchée. Le principe de la subdivision locale est illustré sur la figure 4.8.

Chaque triangle non-représentatif vient à former trois nouveaux triangles grâce à l'ajout d'un neurone supplémentaire au centre du triangle et à l'apparition de 3 nouvelles liaisons de voisinage topologique entre les neurones.

Le choix de cette méthode, dite de subdivision locale, permet de garder une structure de maillage triangulaire cohérente sur la totalité de l'architecture du réseau 3D-SOM. Cette méthode est aussi d'une grande simplicité algorithmique.

5.3. Apprentissage des zones subdivisées et des zones non subdivisées

La procédure de subdivision et d'adaptation locale consiste à appliquer un second processus d'adaptation à la suite du processus d'apprentissage initial.

Après identification des triangles non-représentatifs et application de la procédure de subdivision locale, les neurones appartenant aux triangles subdivisés vont subir une loi d'adaptation différente de celle appliquée au reste des neurones du maillage du réseau 3D-SOM. Cette nouvelle loi d'adaptation utilise localement un plus grand coefficient d'apprentissage que celui appliqué aux zones non subdivisées du maillage, afin de permettre aux neurones appartenant aux triangles non-représentatifs de se rapprocher efficacement des points d'entrée appartenant à ces zones d'adaptation.

Pour les neurones restants du maillage, c'est-à-dire pour ceux qui n'appartiennent pas à un triangle non-représentatif ayant subi une subdivision, l'adaptation suit son cours normal, parallèlement à celle des zones subdivisées, de façon à permettre une adaptation cohérente à la frontière des zones subdivisées et des zones non subdivisées.

Ainsi, les lois d'adaptation appliquées aux neurones en fonction de la nature du triangle auquel ils appartiennent sont les suivantes :

- Si le neurone appartient au moins à un triangle subdivisé :

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha_1(t)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) \leq N_c(t) \\ m_i(t); & \text{si } d_v(c,i) > N_c(t) \end{cases} \quad (4.1)$$

- Si le neurone n'appartient à aucun triangle subdivisé :

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha_2(t)h_{ci}(t)[x(t) - m_i(t)]; & \text{si } d_v(c,i) \leq N_c(t) \\ m_i(t); & \text{si } d_v(c,i) > N_c(t) \end{cases} \quad (4.2)$$

Dans ce contexte d'apprentissage différencié, chaque neurone finit par suivre son propre processus d'adaptation, avec des paramètres d'apprentissage et un voisinage d'adaptation différents suivant la nature du triangle auquel il appartient. Ainsi, dans l'équation (4.1), le coefficient d'apprentissage $\alpha_1(t)$ appliqué aux neurones appartenant aux triangles subdivisés est plus grand que le coefficient $\alpha_2(t)$ appliqué aux neurones appartenant à des triangles non-subdivisés (équation (4.2)). En ce qui concerne le voisinage d'adaptation, si un neurone appartient à un triangle subdivisé et qu'il est dans le voisinage d'adaptation d'un neurone

gagnant qui n'est pas issu d'un triangle subdivisé, il suivra tout de même la loi d'apprentissage (4.1). A l'inverse, si un neurone appartenant à un triangle subdivisé est élu vainqueur et que son voisinage d'adaptation contient des neurones ne faisant pas partie de triangles subdivisés, ces derniers suivront la loi d'adaptation (4.2) (cf. figure 4.9).

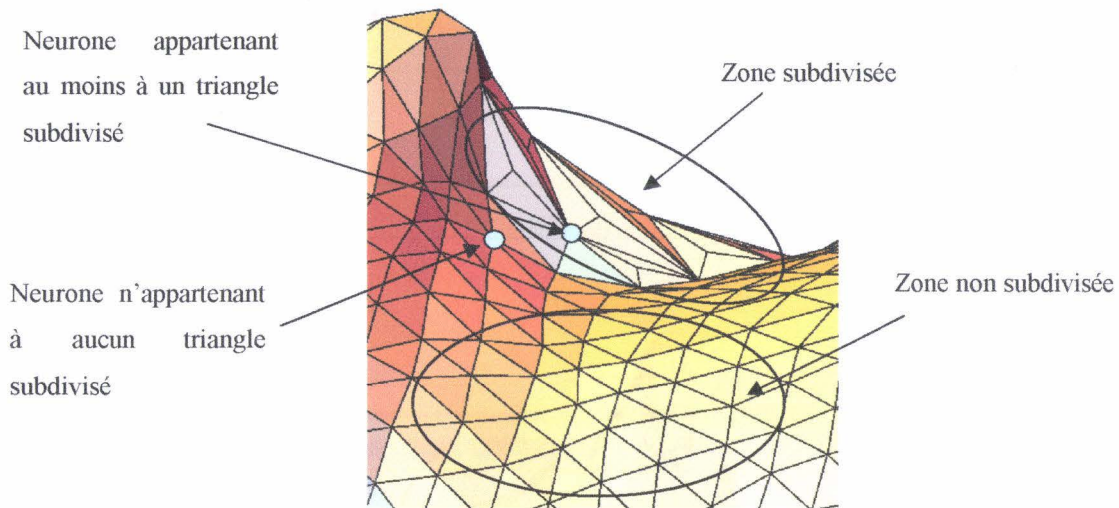
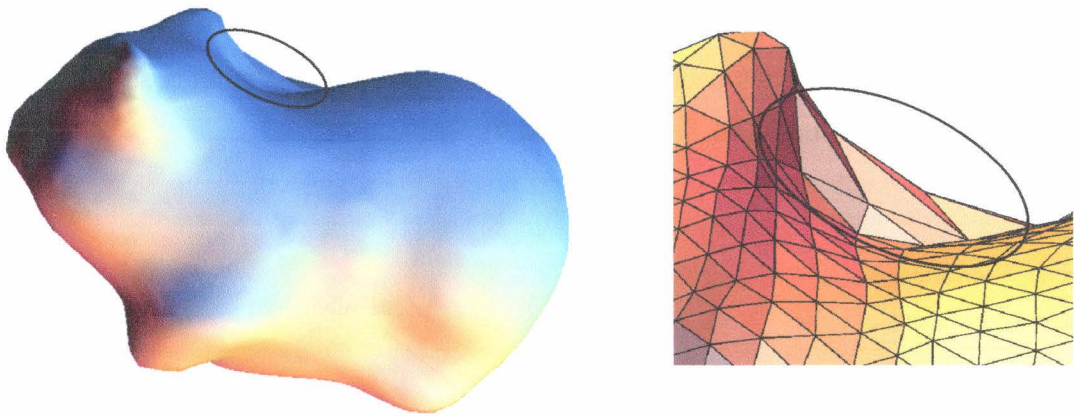


Figure 4.9 Zone subdivisée et zone normale

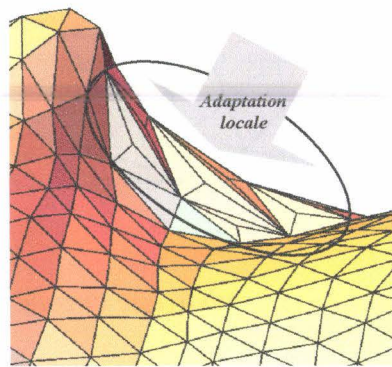
La procédure de subdivision et d'adaptation est relativement rapide du fait qu'au moment de sa mise en oeuvre, le maillage est déjà partiellement formé et donc globalement déjà représentatif de la forme de l'objet à reconstruire. En effet, la majorité des neurones représentent déjà une grande partie du nuage de points en entrée et les paramètres d'adaptation ont déjà des valeurs réduites.

Une illustration de cette méthode est présentée sur la figure 4.10. Sur le détail de la figure 4.10a nous pouvons remarquer une zone de mauvaise adaptation entre l'oreille de la figurine et son dos. Cette zone correspond à une zone à forte concavité de la surface. La prise en compte de ce défaut, puis son élimination s'effectue d'abord avec une identification des triangles non-représentatifs, puis avec l'application du processus de subdivision locale (cf. figure 4.10b).

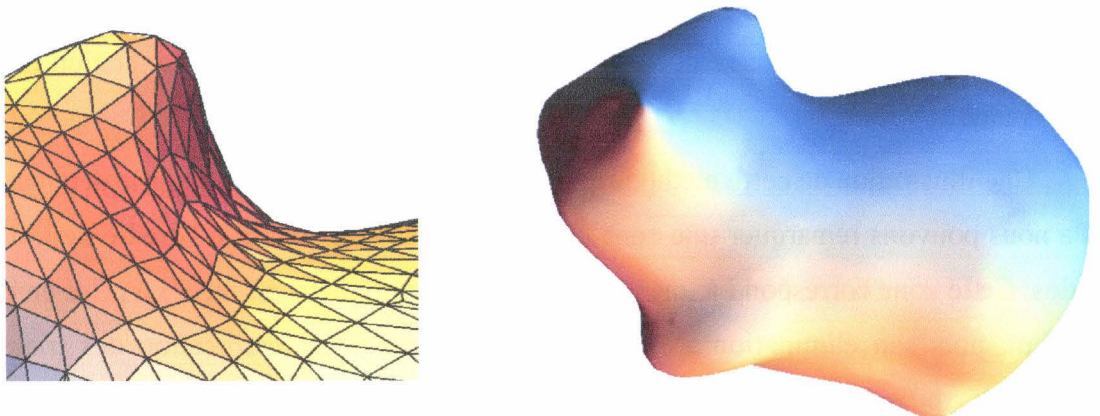
Sur la figure 4.10c nous constatons que le problème a été résolu après application du procédé d'adaptation. Le procédé d'identification des triangles non-représentatifs suivi de la procédure d'adaptation locale se sont montrés efficaces. Remarquons également que le procédé d'auto-organisation conserve l'avantage d'être non-supervisé.



a- Identification des triangles non-représentatifs



b- Subdivision locale et adaptation locale dans la zone subdivisée



c- Résultats

Figure 4.10 Processus de subdivision locale/adaptation locale

6. Prise en compte d'un critère d'orientation de la normale à la surface pour chaque triangle

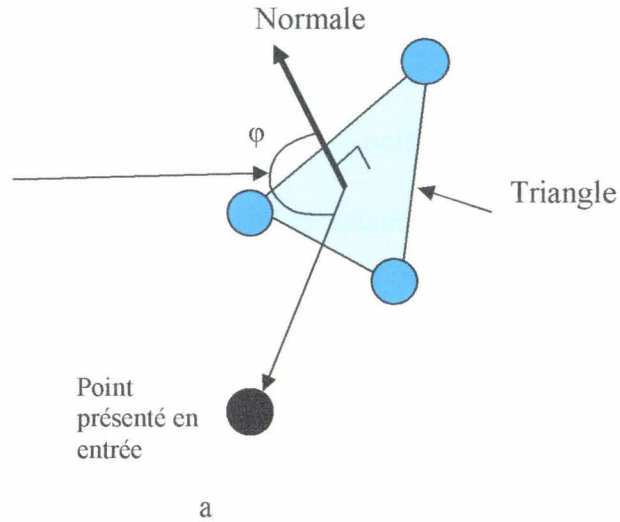
6.1. Intérêt du critère d'orientation

Pour l'ensemble des procédures et lois d'adaptation que nous avons présentées jusqu'à présent, seul le critère de distance, calculé entre les neurones du maillage et le point présenté en entrée est utilisé pour définir le neurone vainqueur. Même si cette approche se révèle efficace dans la plupart des cas, une amélioration de ce critère pourrait fournir de meilleurs résultats pour des nuages de points d'objets 3D présentant des zones à forte concavité et provoquant donc un risque d'apparition de défauts d'adaptation. C'est dans ce contexte que nous proposons d'améliorer le critère permettant d'élire le neurone gagnant en y intégrant la notion d'orientation de la normale à la surface du triangle adapté.

Dans cette approche, l'adaptation des neurones dans le voisinage du neurone vainqueur se fait au niveau de chaque triangle, même si la première phase de l'algorithme, qui consiste à trouver le neurone vainqueur et son voisinage, reste inchangée. La notion de normale à la surface est définie à partir des triangles du maillage. Il est donc naturel d'utiliser le triangle dans son ensemble et non plus le seul neurone comme entité d'adaptation.

Le nouveau critère, que nous appelons « critère d'orientation », prend en compte la direction de la normale à la surface de chaque triangle qui participe à l'apprentissage. Il prend également en compte l'orientation du plan du triangle par rapport au point présenté en entrée du réseau. Pour cela, on considère l'angle non orienté φ entre le vecteur normal à la surface du triangle et le vecteur ayant pour origine le centre de gravité du triangle et orienté vers le point présenté en entrée (cf. figure 4.11a). Dans le voisinage d'adaptation du neurone gagnant, plus cet angle est petit, plus les neurones de ce triangle sont adaptés. A l'inverse, plus cet angle est grand, moins les neurones de ce triangle sont sujets à l'adaptation (cf. figure 4.11b).

L'angle φ représente l'écart angulaire entre le vecteur dont l'origine est le centre de gravité du triangle et l'extrémité le points en entrée et le vecteur normal à la surface de ce triangle



Du fait que $\varphi_2 > \varphi_1$: les neurones du triangle 1 seront plus sujets à adaptation que les neurones du triangle 2

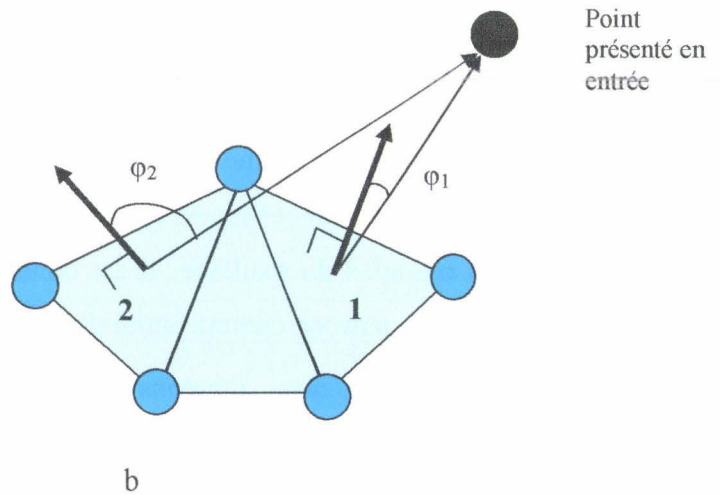


Figure 4.11 Principe de la prise en compte du critère d'orientation de la normale dans le processus d'apprentissage

Le critère d'orientation doit permettre d'assurer une plus grande cohérence d'adaptation pour les neurones du réseau 3D-SOM qui participent à l'apprentissage du nuage de points présenté en entrée. Il permet en effet d'éviter le phénomène d'adaptation où un même neurone se retrouve presque systématiquement, attiré de façon alternée vers des zones distinctes et éloignées. De plus, grâce au critère d'orientation, les neurones qui appartiennent à un même triangle sont nécessairement attirés vers une même zone de la surface de l'objet. Cela limite donc considérablement la possibilité de voir apparaître un étirement important des triangles en cours d'apprentissage.

6.2 Implantation du critère d'orientation dans l'algorithme des 3D-SOM

Le critère d'orientation est appliqué à la procédure d'adaptation des 3D-SOM et permet de réaliser une adaptation modulée des neurones en fonction d'un seuil β appelé seuil d'adaptation. La valeur de ce seuil, définie dans l'intervalle $[0, \pi]$, permet de décider de réaliser ou non l'adaptation des neurones au niveau de chaque triangle en fonction de la valeur de l'angle non orienté ϕ .

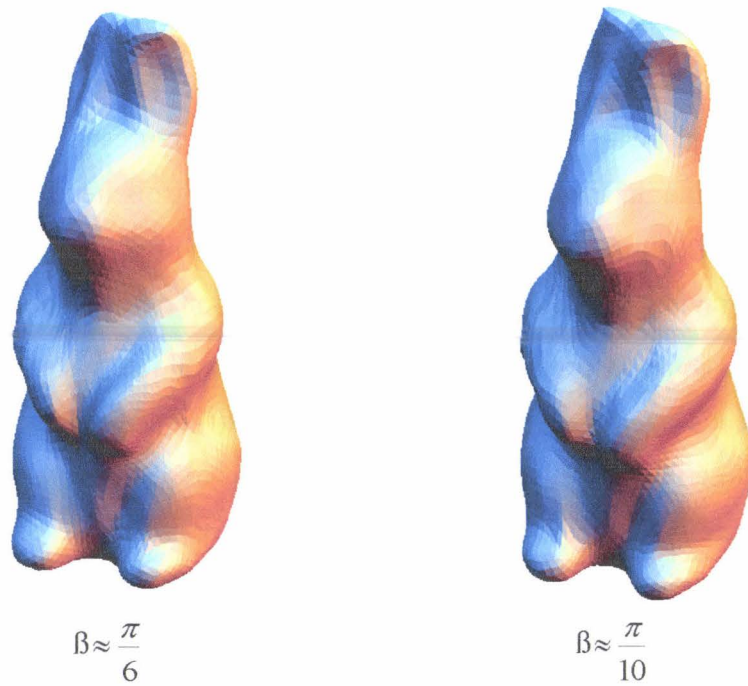
Pour calculer et mettre en œuvre le nouveau critère d'orientation dans la procédure d'adaptation, nous définissons préalablement T comme étant l'ensemble composé de tous les triangles du maillage neuronal avec $T = \{T_1, \dots, T_j, \dots, T_K\}$, où K représente le nombre de triangles du maillage. T_j représente un triangle quelconque composé de trois neurones. Nous notons η_j le vecteur définissant l'orientation normale à la surface du triangle T_j et γ_j le vecteur ayant pour origine le centre de gravité du triangle et orienté vers la position du point présenté en entrée.

De façon classique, la procédure d'apprentissage implique systématiquement le choix dans la base d'apprentissage (nuage de points) d'un point $x(t)$ à présenter en entrée du réseau et la recherche, à l'intérieur de l'architecture du réseau 3D-SOM, du neurone vainqueur correspondant. Ensuite, pour chaque triangle $T_j(t)$ dans le voisinage d'adaptation du neurone gagnant, les vecteurs-poids des neurones qui le compose sont adaptés de façon modulée en fonction de la valeur de l'angle ϕ par rapport au seuil d'adaptation β . Ainsi, pour un point $x(t)$ présenté en entrée du réseau à l'itération de rang t , la procédure d'adaptation est redéfinie comme suit :

- calcul du centre de gravité et de la normale $\eta_j(t)$ à la surface pour le triangle $T_j(t)$.
- calcul du vecteur $\gamma(t)$ dont l'origine est le centre de gravité du triangle $T_j(t)$ et l'extrémité le point en entrée $x(t)$.
- calcul de l'angle non orienté ϕ : $\phi = \arccos \left(\frac{\gamma(t) \cdot \eta_j(t)}{\|\gamma(t)\| \|\eta_j(t)\|} \right)$ avec $\phi \in [0, \pi]$.
- application de l'adaptation en fonction de la valeur du seuil β

$$m_i(t+1) = \begin{cases} m_i(t) + h_{ci}(t)\alpha(t)(x(t) - m_i(t)); & \text{si } d_v(c,i) \leq N_c(t) \text{ et } \phi \leq \beta \\ m_i(t) + h_{ci}(t)(\alpha(t)/\chi)(x(t) - m_i(t)); & \text{si } d_v(c,i) \leq N_c(t) \text{ et } \phi > \beta \\ m_i(t); & \text{si } d_v(c,i) > N_c(t) \end{cases}$$

où χ représente un coefficient de pondération qui limite le coefficient d'adaptation des neurones situés dans les zones ne satisfaisant pas au critère $\varphi \leq \beta$, mais qui font néanmoins partie du voisinage d'adaptation du neurone gagnant. Pour l'exemple de la figure 4.12 nous avons choisi la valeur 100 pour le paramètre χ .



Deux exemples qui nous renseignent sur la viabilité de la méthode permettant la prise en compte du critère d'orientation

Figure 4.12 Exemples de reconstruction d'une surface 3D pour 2 valeurs différentes du seuil β

L'inconvénient lié à cette méthode est la répétition des adaptations de chaque neurone en fonction du nombre de triangles auxquels il appartient. Néanmoins, cette répétition des adaptations permet d'avoir un renforcement local de l'adaptation de chaque neurone. La surcharge de calculs est limitée par le fait qu'à chaque pas d'apprentissage, le nombre de neurones qui participent à l'adaptation diminue. De plus, le voisinage d'adaptation en début de procédure est déjà réduit car la procédure est appliquée à une structure neuronale déjà préformée.

La figure 4.12 illustre, au travers d'un exemple de reconstruction, l'effet de la procédure d'adaptation avec critère d'orientation. Le résultat obtenu sur la figure 4.12 est très satisfaisant au niveau de la qualité de la représentation, ce qui montre l'intérêt de cette nouvelle approche.

7. Exemples de reconstructions de surfaces par 3D-SOM

Nous nous proposons ici de présenter quelques résultats de reconstruction d'objets 3D afin de montrer l'efficacité du réseau 3D-SOM sur des exemples synthétiques et réels de complexités diverses. Les deux exemples suivants ont été reconstruits en utilisant l'algorithme prenant en compte la notion de densité de points.

Un premier exemple consiste en un nuage de points représentant une dent (cf. figure 4.13a). L'objet a été reconstruit en utilisant une carte composée de 6402 neurones (cf. figure 4.13b). Le nombre d'itérations nécessaire pour atteindre le résultat présenté sur la figure 4.13b est de 12000.

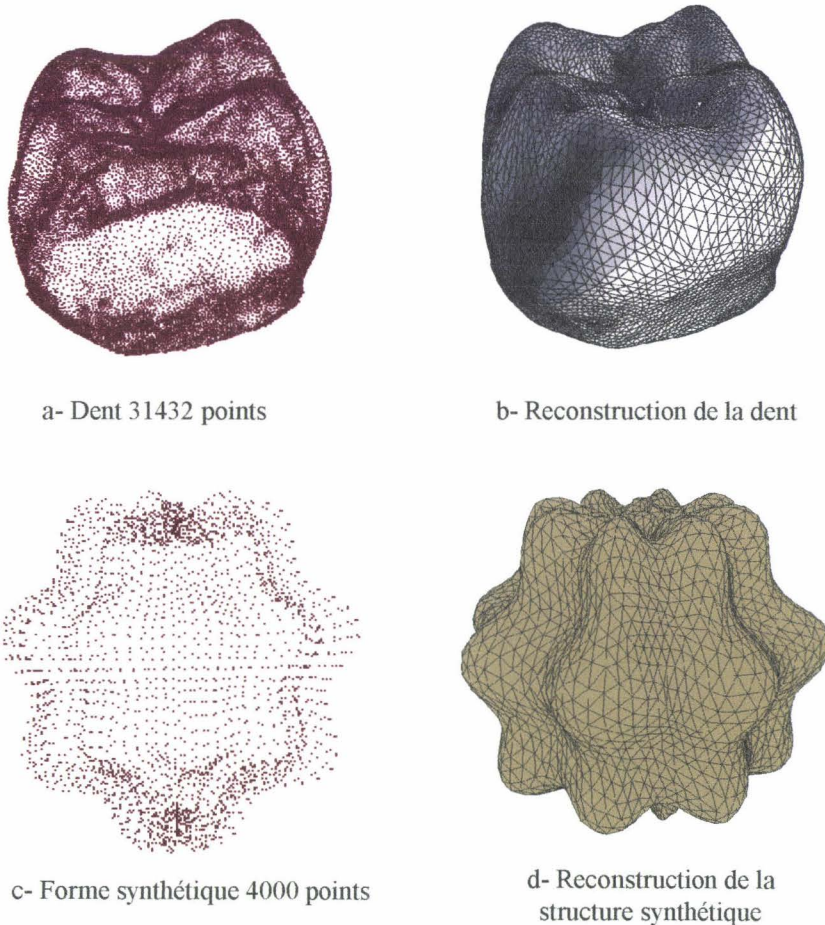


Figure 4.13 Exemples d'objets 3D (dent et forme synthétique) représentés par leurs nuages de points et leurs reconstructions

Le second exemple consiste en un nuage de points représentant une structure synthétique (cf. figure 4.13c). L'objet 3D a été reconstruit en utilisant une carte composée de 3568 neurones. Le nombre d'itérations nécessaire pour atteindre le résultat présenté sur la figure 4.13d est de 10000.

Le nombre d'itérations pour arriver à ces résultats sont relativement comparables, ce qui tend à montrer l'influence limitée de la taille du nuage de points en entrée sur l'apprentissage.

Seulement quelques minutes sont nécessaires, même pour de grands maillages, pour reconstruire la surface finale. Par exemple, le temps nécessaire pour obtenir la forme de la dent est approximativement de 6 minutes sur une machine avec un processeur Pentium 4, équipée de 512MB de RAM et d'une carte graphique 256MB.

La qualité et le nombre d'itérations pour obtenir ces résultats sont fortement liés au choix des paramètres de l'algorithme que sont le niveau du voisinage d'adaptation $N_c(0)$ et son coefficient de décroissance d , le coefficient d'apprentissage $\alpha(0)$ et également son exposant de décroissance D . Un coefficient d'apprentissage qui diminue trop rapidement ou un trop petit voisinage d'adaptation mèneraient à une carte difforme ou à d'importants défauts d'adaptation.

Nous présentons dans ce paragraphe, des exemples variés de reconstruction de surfaces utilisant les 3D-SOM. Les objets utilisés concernent des os humains, des statuettes, un buste de Max Planck, etc.. La figure 4.14 regroupe les reconstructions des nuages de points présentées dans le tableau 4.1. Ce tableau est un récapitulatif des caractéristiques des différents nuages de points utilisés dans nos exemples avec, pour chaque nuage, son origine et le nombre de points le constituant.

<i>Exemples</i>	<i>Origine</i>	<i>Nombre de points</i>
Buste de déesse grecque (cf. figure 4.14a)	http://www.cyberware.com	50003
Fémur (cf. figure 4.14b)	http://www.cyberware.com	137062
Max Planck (cf. figure 4.14c)	Institut Max Planck	70072
Humérus (cf. figure 4.14d)	http://www.ai.cluny.ensam.fr/	138966
Ecureuil (cf. figure 4.14e)	http://www.mpisb.mpg.de/~shin/	9995
Pied (cf. figure 4.14f)	http://research.microsoft.com/~hoppe/	20021
Crâne (cf. figure 4.14g)	http://www.headus.com	32000
Dentier (cf. figure 4.14h)	http://www.cyberware.com	50000

Tableau 4.1 Caractéristiques des nuages de points ayant servis pour les exemples de reconstruction 3D de la figure 4.14

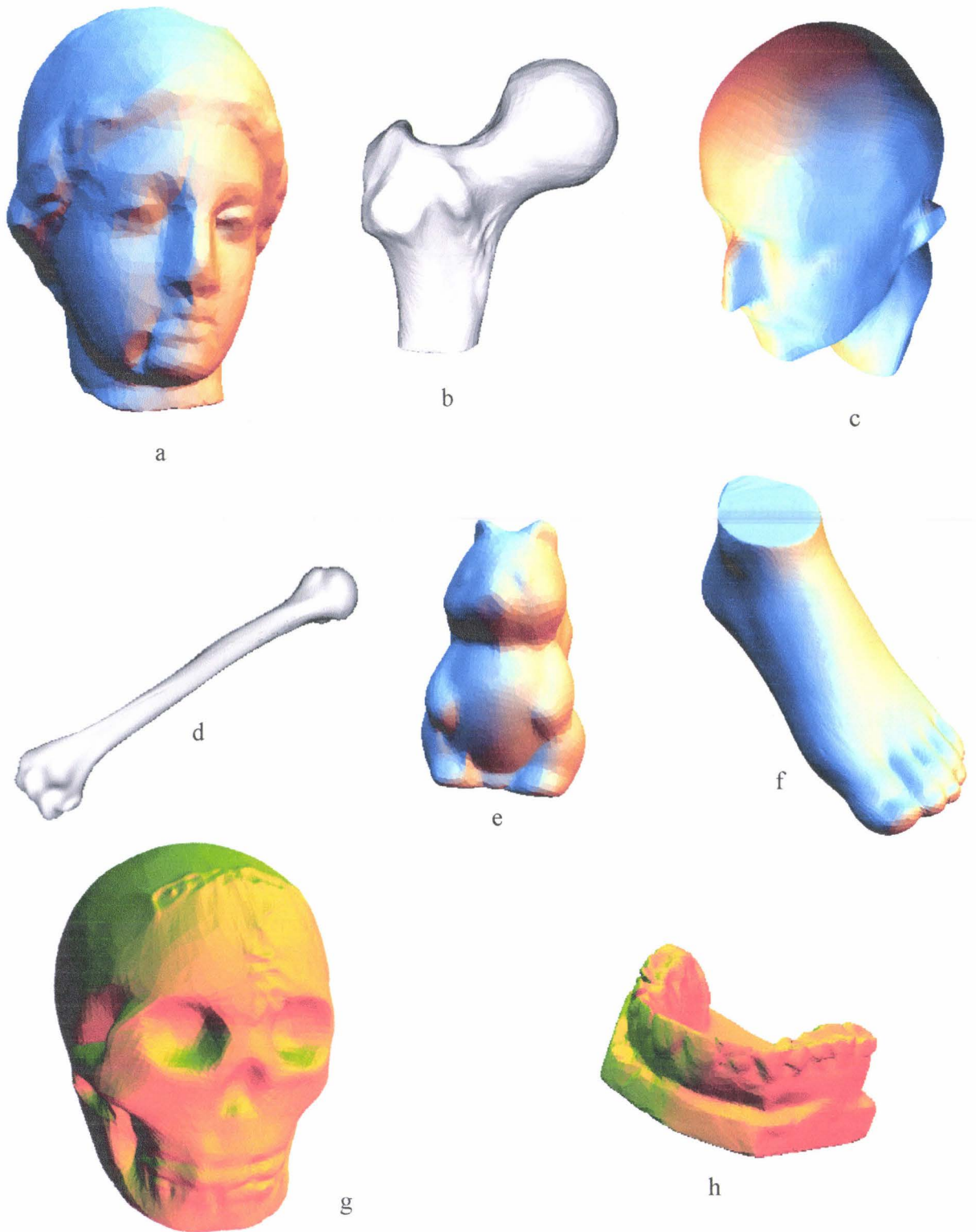


Figure 4.14 Résultats de reconstruction

8. Conclusion

Dans ce chapitre plusieurs améliorations du processus de reconstruction obtenues à l'aide des 3D-SOM ont été présentées. Nous avons tout d'abord exposé une méthode qui tient compte de la densité locale des points du nuage de points présenté en entrée du réseau, puis une méthode de subdivision et d'adaptation locale et enfin une méthode prenant en compte l'orientation de la normale à la surface au centre de chaque triangle du maillage. L'ensemble de ces méthodes a pour but d'améliorer la qualité et la fidélité de représentation des détails surfaciques présents sur les objets reconstruits.

Néanmoins, vu le caractère novateur de ces méthodes, ne négligeons pas le fait qu'il reste sans doute encore des améliorations possibles, que ce soit au niveau de la qualité de représentation ou des méthodes d'adaptation.

Conclusion générale

1. Conclusion

Dans ce mémoire, nous nous sommes intéressés à l'application des cartes auto-organisatrices au domaine de la reconstruction de surfaces à partir de nuages de points non-organisés. Nous présentons une nouvelle architecture neuronale compétitive, appelée 3D-SOM, dévolue à la représentation structurée de nuages de points acquis à la surface d'objets 3D.

L'architecture des 3D-SOM offre l'avantage de permettre la généralisation de la notion d'architecture topologique des cartes auto-organisatrices. Le fait de partir d'une structure complètement désorganisée pour arriver à une représentation de la forme à modéliser, nous renseigne sur les qualités d'adaptation des 3D-SOM. L'utilisation des réseaux 3D-SOM sur différents exemples d'adaptation a ainsi permis d'obtenir des résultats de reconstruction très encourageants.

Nous avons aussi apporté de nouvelles solutions pour améliorer la qualité de représentation des 3D-SOM. La prise en compte de la densité dans le nuage de points, les stratégies de réadaptation, de subdivision, d'adaptation locale et d'apprentissage prenant en compte la notion d'orientation des normales à la surface du maillage neuronal ont conduit à des résultats exempts de défauts apparents.

2. Perspectives

Dans la perspective de la poursuite de ce travail de recherche, nous avons réfléchi à quelques améliorations qui pourraient rendre le processus de reconstruction plus efficace.

Les méthodes de subdivision et d'arrangement de l'espace 3D tels que les arbres octenaires ou octrees pourraient être employées pour accélérer la recherche des neurones gagnants.

Pour résoudre le problème des défauts d'adaptation nous avons testé une solution se basant sur une subdivision locale du maillage neuronal et sur une adaptation locale des zones subdivisées. Une méthode d'élimination des triangles problématiques pourrait être mise en place, sans chercher à corriger le problème par l'apprentissage mais en le supprimant. Cela impliquerait une procédure en aval de re-maillage de la carte 3D-SOM.

Cette méthode de re-maillage pourrait servir de base pour un apprentissage autonome de la carte, surtout dans le cas d'objets de topologie complexe, par exemple des surfaces comptant de nombreux trous.

La comparaison de notre méthode avec d'autres approches non neuronales n'est pas triviale, mais nous espérons avoir su apporter et présenter des améliorations dans la qualité et la simplicité de la reconstruction 3D et de sa représentation par réseaux de neurones à partir d'un nuage de points.

Références personnelles

Farid Boudjemaï, Philippe Biela Enberg, Jack-Gérard Postaire, "Surface modeling by using self-organizing maps of KOHONEN", **IEEE International Conference on Systems, Man and Cybernetics, SMC'03**, Washington D.C, vol.3 pp. 2418 – 2423.

Farid Boudjemaï, Philippe Biela Enberg, Jack-Gérard Postaire, "Self Organizing Spherical Map Architecture for 3D Object Modeling", **2003 Workshop on Self-Organizing Maps, WSOM'03**, Hibikino, Kitakyushu, Japan, pp. 197- 202.

Farid Boudjemaï, Philippe Biela Enberg, Jack-Gérard Postaire, "Réseaux de neurones à architectures évolutives pour la modélisation de surfaces 3D", **Conférence Internationale Francophone d'Automatique, CIFA'04**, Douz, Tunisia.

Farid Boudjemaï, Philippe Biela Enberg, Jack-Gérard Postaire, "3D Self Organizing Convex Neural Network Architectures ", **International Conference on Adaptive and Natural Computing Algorithms, ICANNGA'05**, Coimbra, Portugal. Springer Wien-New York, pp. 164-167.

Farid Boudjemaï, Philippe Biela Enberg, Jack-Gérard Postaire, "Dynamic adaptation and subdivision in 3D-SOM: application to surface reconstruction", **IEEE International Conference on Tools with Artificial Intelligence, ICTAI'05**, Hong Kong, China.

Bibliographie

- [All98] Allgower E.L. and Schmidt P.H. (1998). An algorithm for piecewise linear approximation of an implicitly defined manifold, *SIAM Journal of Numerical Analysis*, 22:322-346.
- [Ama79] Amari S. and Takeuchi A. (1979). Formation of topographic maps and columnar microstructures in nerve fields, *Biological Cybernetics*, 35:36-72.
- [Ame98] Amenta N., Bern M. and Kamvysselis M. (1998). A new Voronoi-based surface reconstruction algorithm, *Siggraph '98*, 415-421.
- [Ame99] Amenta N., Bern M. and Choi C. (1999). One-Pass Delaunay filtering for homeomorphic 3D surface reconstruction, *UT Technical report number TR99-08*.
- [Ame01] Amenta N., Choi C. and Kolluri R. (2001). The Power Crust, *Proceedings of 6th ACM Symposium on Solid Modeling*, 249-260.
- [Baa93] Baader A. and Hirzinger G. (1993). Three-dimensional surface reconstruction based on a Self-organizing feature map, In *Proc. 6th Int. Conf. Advan. Robotics*, 273-278.
- [Baa94] Baader A. and Hirzinger G. (1994). A Self-organizing algorithm for multisensory surface reconstruction, In *International Conf. on Robotics and Intelligent Systems IROS'94*, Munich, Germany.
- [Bar95] Bardinet E., Cohen L.D. and Ayache N. (1995). A parametric deformable model to fit unstructured 3D data, *Rapport de recherche de l'INRIA - Sophia Antipolis*, Equipe : EPIDAURE.
- [Bar01] Barhak J. and Fischer A. (2001). Adaptive reconstruction of freeform objects with 3D SOM neural network grids, *Ninth Pacific Conference on Computer Graphics and Applications*.

- [Bit95] Bittar E., Tsingos N. and Gascuel M.P. (1995). Automatic reconstruction of unstructured 3D data : combining medial axis and implicit surfaces, Computer Graphics Forum.
- [Bie99] Biela P. (1999). Classification automatique d'observation multidimensionnelles par réseaux de neurones compétitifs, PhD thesis, Université des Sciences et Technologies de Lille.
- [Bli82] Blinn J.E. (1982). Light reflection functions for simulation of clouds and dusty surfaces, Computer Graphics, 3:21-29.
- [Blo97] Bloomenthal J. (1997). Introduction to implicit surfaces, Morgan Kaufmann publishers Inc.
- [Blu67] Blum H. (1967). A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form (Walthen-Dunn, W., ed.) MIT Press.
- [Boi84] Boissonnat J.D. (1984). Geometric structures for three-dimensional shape representation, ACM Trans. Graph., 3(4):266-286.
- [Boi00] Boissonnat J.D. and Cazals F. (2000). Smooth surface reconstruction via natural neighbour interpolation of distance functions, In Proceedings of the sixteenth annual symposium on Computational geometry, 223-232.
- [Bou93] Bouton C. and Pages G. (1993). Self-organization of the one-dimensional Kohonen algorithm with non-uniformly distributed stimuli, Stochastic Processes and their Applications, 47:249-274.
- [Bru90] Brunak S. and Lautrup B. (1990). Neural Networks, computers with intuition, World Scientific, Singapore.
- [Car87a] Carpenter G.A. and Grossberg S. (1987). Invariant pattern recognition and recall by an attentive Self-organization ART architecture in a nonstationary World, IEEE ICNN(San Diego), 2:737-746.
- [Car87b] Carpenter G.A. and Grossberg S. (1987). ART2: Stable Self-organization of category recognition codes for analog input patterns, Applied Optics, 26:4919-4930.

- [Car90] Carpenter G.A. and Grossberg S. (1990). ART3: Hierarchical search using chemical transmitters in Self-organizing pattern recognition architecture, *Neural Networks*, 3:129-152.
- [Caz04] Cazals F. and Giesen J. (2004). Delaunay Triangulation based Surface Reconstruction: Idas and Algorithms, INRIA Research Report 5393.
- [Che03] Chen D., Chang R. and Huang Y. (2003). Breast cancer diagnosis using Self-organizing maps for sonography, *Ultrasound in Medicine and Biology*, 26:405-411.
- [Cot87] Cottrell M. and Fort J.C. (1987). Étude d'un processus d'auto-organisation. *Annales de l'Institut Henri Poincaré*, 23(1):1-20.
- [Cot94] Cottrell M., Fort J.C. and Pagès G. (1994). Two or three things that we know about the Kohonen algorithm, *Proc. ESANN'94, European Symp. on Artificial Neural Networks*, Brussels, 235-244.
- [Dey99] Dey K. and Kumar P. (1999). A simple provable algorithm for curve reconstruction, *Proc. 10th. ACM-SIAM Symposium on Discrete Algorithm*, 893-894.
- [Dey01] Dey T.K., Giesen J. and Hudson J. (2001). Delaunay Based shape reconstruction from large data, *In proceedings of IEEE Visualization*.
- [Die97] Diestel R. (1997). *Graph Theory*, Springer.
- [Ekl03] Eklund T., Back B., Vanharanta H. and Visa A. (2003). Financial benchmarking using Self-organizing maps, studying the international pulp and paper industry Source Data mining: opportunities and challenges, 323-349.
- [Erw92a] Erwin E., Obermayer K., and Schulten K. (1992a). Self-organizing maps: Ordering, convergence properties and energy functions. *Biological Cybernetics*, 67(1):47-55.
- [Erw92b] Erwin E., Obermayer K., and Schulten K. (1992b). Self-organizing maps: Stationary states, metastability and convergence rate. *Biological Cybernetics*, 67(1):35-45.

- [For93] Fort J.C. and Pagès G. (1993). Sur la convergence p.s. de l'algorithme de Kohonen généralisé, compte rendu de l'Académie des Sciences, série I, 317 :389-394.
- [Fra80] Franke R. and Nielson G. (1980). Smooth interpolation of large sets of scattered data, International Journal for Numerical Methods in Engineering, 15(11) :1691-1704.
- [Fri95] Fritzke B. (1995). A Growing Neural Gas Network learns topologies, Advances in Neural Information Processing Systems 7, MIT Press, 625-632.
- [Fuk75] Fukushima K. (1975). Cognitron: A Self-organizing multilayered Neural Network, Biological Cybernetics, 20:121-136.
- [Gao03] Gao S. and Lu H.Q. (2003). A fast algorithm for delaunay based reconstruction, In Proc. 11th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG2003).
- [Gop00] Gopi M., Krishnan S. and Silva C.T. (2000). Surface reconstruction based on lower dimensional localized delaunay triangulation, In Proceedings of Eurographics, 19:C476-C478,
- [Gou00] Goulette F. (2000). Etat de l'art des capteurs de numérisation 3d, Scanning 2000, Paris, France.
- [Gro76] Grossberg S. (1976). Adaptive pattern classification and universal recoding, feedback, expectation, olfaction, and illusions, Biological Cybernetics, 23:187-202.
- [Gro88] Grossberg S. (1988). Nonlinear Neural Networks: principles, mechanisms, and architectures, Neural Networks, 1:17-61.
- [Har93] Hart J. (1993). Ray tracing implicit surfaces, Siggraph 93 Course Notes: Design, Visualization and Animation of Implicit Surfaces, 1-16.
- [Hay94] Haykin S. (1994) Neural Networks: A comprehensive foundation. McMillan College Publishing Co.

- [He94] He T., Wang S. and Kaufman A. (1994). Wavelet-based volume morphing, *Proceedings of Visualization 94*, 85-92.
- [Hér94] Hérault J. et Jutten C. (1994). *Les réseaux de neurones et le traitement du signal*, Editions Hermès.
- [Hon97] Honkela T. Kaski S. Lagus K. & Kohonen T. (1997). WEBSOM Self-organizing maps of document collections, *Proceedings of WSOM'97: Workshop on Self-organizing Maps*, Espoo, Finland: Helsinki University of Technology, 310-315.
- [Hop82] Hopfield J.J. (1982). Neural Networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences, USA*, 79:2554-2558.
- [Hop92] Hoppe H., DeRose T., Duchamp T., McDonald J. and Stuetzle W. (1992). Surface reconstruction from unorganized points, *Computer Graphics*, 26(2) :71-78.
- [Hop93] Hoppe H., DeRose T., Duchamp T., McDonald J. and Stuetzle W. (1993). Mesh optimization, *Computer Graphics*, 27:19-26.
- [Hop94] Hoppe H., DeRose T., Duchamp T., Halstead M., Jin H., McDonald J., Schweitzer J. and Stuetzle W. (1994). Piecewise smooth surface reconstruction, *Computer Graphics*, 28:295-302.
- [Hub77] Hubel D. and Wiesel T. (1977). Ferrier lecture: Functional architecture of macaque monkey visual cortex. *Proc. Roy. Soc. Lond. B. Biol. Sci.*, 198:1-59.
- [Hug92] Hughes J.F. (1992). Scheduled fourier volume morphing, *SIGGRAPH*.
- [Ivr03] Ivriissimtzis I., Jeong W.K. and Seidel H.S. (2003) Using growing cell structures for surface reconstruction, *Shape Modeling International (SMI 2003)*, Seoul, Korea, IEEE, Los Alamitos, 78-86.
- [Kas87] Kass M., Witkin A. and Terzopoulos D. (1987). Snakes : Active contour models, *Computer Vision*, 1:321-331.

- [Kas95] Kaski S. and Kohonen T. (1995) Exploratory data analysis by the Self-organizing map: structures of welfare and poverty in the world, *Proceedings of Neural Networks in the Capital Markets*, World Scientific, Singapore, 498-507.
- [Koh81a] Kohonen T. (1981a). Automatic formation of topological maps of patterns in a Self-organizing system, In Oja, E. and Simula, O., editors, *Proceedings of 2SCIA, Scand. Conference on Image Analysis*, Helsinki, Finland. Suomen Hahmontunnistustutkimuksen Seura r.y., 214-220.
- [Koh81b] Kohonen T. (1981b). Construction of similarity diagrams for phonemes by a Self-organizing algorithm, Technical Report TKK-F-A463, Helsinki University of Technology, Espoo, Finland.
- [Koh81c] Kohonen T. (1981c). Hierarchical ordering of vectorial data in a Self-organizing process, Report TKK-F-A461, Helsinki University of Technology, Espoo, Finland.
- [Koh81d] Kohonen T. (1981d). Self-organized formation of generalized topological maps of observations in a physical system, Report TKK-F-A450, Helsinki University of Technology, Espoo, Finland.
- [Koh82a] Kohonen T. (1982a). Analysis of a simple Self-organizing process, *Biological Cybernetics*, 44(2):135-140.
- [Koh82b] Kohonen T. (1982b). Self-organizing formation of topologically correct feature maps, *Biological Cybernetics*, 43(1):59-69.
- [Koh84] Kohonen T. (1984). *Self-organization and Associative Memory*, Springer-Verlag, New York.
- [Koh88] Kohonen T. (1988). The 'neural' phonetic typewriter, *Computer*, 21(3):11-22.
- [Koh01] Kohonen T. (2001). *Self-organizing maps*, Springer, Berlin, Third Edition, Heidelberg.
- [Knu79] Knudsen E. and Konishi M. (1979). Mechanisms of sound localization in the barn owl (*tyto alba*), *Journal of Comparative Physiology*, 133:13-21.
- [Lip87] Lippman R. (1987). An introduction to computing with neural nets, *IEEE ASSP Magazine*, 4-22.

- [Lo91] Lo Z.P., Bavarian B. (1991). On the rate of convergence in topology preserving Neural Networks, *Biological Cybernetics*, 65(1):55-63,
- [Lo93] Lo Z.P., Yu Y. and Bavarian B. (1993). Analysis of the convergence properties of topology preserving Neural Networks, *IEEE Transactions on Neural Networks*, 4: 207-220.
- [Mal73] Von der Malsburg C. (1973). Self-organization of orientation sensitive cells in the striate cortex, *Kybernetik*, 14:85-100.
- [Mar91] Martinetz T.M., Ritter H. J. and Schulten K. J. (1991). *Neuronale Netze*, Addison-Wesley, München.
- [McC43] McCullough W. S. and Pitts W. (1943) A logical calculus of ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5: 115-133.
- [Min69] Minsky M. and Papert, S. (1969). *Perceptrons*, published by MIT Press.
- [Mur91] Muraki S. (1991). Volumetric shape description of range data using blobby model, *SIGGRAPH 1991*, 227-235.
- [Pen50] Penfield W. and Rassmussen T. (1950). *The cerebral cortex of man*, Macmillan.
- [Oth03] Ohtake Y., Belyaev A., Alexa M., Turk G. and Seidel H.P. (2003). Multi-level Partition of Unity Implicits, *SIGGRAPH 2003*, San Diego, California USA, 27-31
- [Qin01] Qin H. and Duan Y. (2001). Intelligent Balloon: a subdivision-based deformable model for surface reconstruction of arbitrary, Unknown Topology, In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, 47-58.
- [Reu03] Reuter P. (2003). *Reconstruction and rendering of implicit Surfaces from large unorganized point sets*, PhD thesis, LaBRI – University of Bordeaux 1.

- [Rit86] Ritter H. and Schulten K. (1986). Kohonen's Self-organizing maps: exploring their computational capabilities, *Biological Cybernetics* 54:99-106.
- [Rit88] Ritter b H. and Schulten K. (1988). Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection, *Biological Cybernetics* 60:59-71.
- [Rit92] Ritter H., Martinetz T. and Schulten K. (1992). *Neural computation and Self-organizing maps: An introduction*, Addison-Wesley, New York, revised English edition.
- [Ros58] Rosenblat F. (1958). The Perceptron: a Probabilistic Model for Information Storage and Organisation in the Brain, *Psychological Review*, 65:386-408.
- [Ros94] Rossignac J. and Kaul A. (1994). Agrepls and bips: metamorphosis as a Bezier curve in the space of polyhedra, *Eurographics*.
- [Rum85] Rumelhart D. and Zipser D. (1985). Feature discovery by competitive learning, *Cognitive Science*, 9:75-112.
- [Rum86] Rumelhart D., Hinton G. and Williams R. (1986). *Learning internal Representations by Error Propagation*, Parallel Distributed Processing, Rumelhart & Mc Clellan, MIT Presss.
- [Sch05] Schall O. and Samozino M. (2005). Surface from Scattered Points: A Brief Survey of Recent Developments, 1st International Workshop on Semantic Virtual Environments, 138-147.
- [Tau93] Taubin G. (1993). An improved algorithm for algebraic curve and surface fitting, *IEEE Proceedings of Computer Vision and Pattern Recognition*, 658-665.
- [Thi92] Thiran P. and Hasler M. (1992). Réseau de Kohonen avec poids synaptiques quantifiés, In M. Cottrell and M. Chaleyat-Maurel editors *Proc. Workshop Aspects Théoriques des Réseaux de Neurones* Paris France Université Paris I.

[Try91] Tryba V. and Goser K. (1991). Self-organizing feature maps for process control in chemistry, In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, Amsterdam, Netherlands, 847-852

[Tur02] Turk G. and O'Brien J.F. (2002). Implicit surfaces that interpolate, In SMI.

[Wil76] Willshaw D. J. and Von der Malsburg C. (1976). How patterned neural connections can be set up by Self-organization, *Proceedings of the Royal Society of London*, B 194:431-445.

[Woo81] Woolsey C. (1981). *Cortical sensory organization: Multiple auditory areas*, Crescent Manor, N.J.: Humana.

[Wyv86] Wyvill G., McPheeters C. and Wyvill B. (1986). Data structures for soft objects, *The Visual Computer*, 2(4):227-234.

[Xer98] Xerri C. (1998). Plasticité post-lésionnelle des cartes corticales somatosensorielles : une revue, *compte rendu de l'Académie des Sciences, Paris*, 321:135-151.

[Yu99] Yu Y. (1999) Surface reconstruction from unorganized points using Self-organizing Neural Networks, In *IEEE Visualization*.

RECONSTRUCTION DE SURFACES D'OBJETS 3D A PARTIR DE NUAGES DE POINTS PAR RESEAUX DE NEURONES 3D-SOM

Résumé :

Les travaux menés dans le cadre de cette thèse ont pour principal objectif de développer une architecture neuronale auto-adaptative et non supervisée dévolue à la reconstruction d'objets 3D. L'espace d'entrée du réseau de neurones est constitué du nuage de points 3D non-organisés acquis à la surface de l'objet à reconstruire. Nos travaux se sont orientés vers le développement d'une nouvelle architecture neuronale inspirée des cartes auto-organisatrices de Kohonen que nous avons appelée 3D-SOM.

Après apprentissage du nuage de points par le réseau, l'architecture du réseau se présente sous la forme d'une architecture neuronale maillée ayant pris la forme de l'objet à reconstruire.

Nous avons adjoint au réseau 3D-SOM différentes lois d'apprentissage, permettant la mise en œuvre d'un processus d'auto-organisation efficace. Nous avons également introduit :

- un niveau d'adaptation modulable qui permet l'évolution dynamique du voisinage d'apprentissage.
- une loi d'adaptation locale subordonnée à des critères spécifiques pour chaque neurone ou pour chaque triangle (coefficient d'apprentissage local, densité locale des points dans le nuage, orientation des normales aux triangles).
- un processus de subdivision d'architecture global ou local qui consiste à accroître le nombre de neurones (donc du nombre de mailles) composant le réseau aux endroits où le réseau rencontre des difficultés à s'adapter à la forme réelle de l'objet.

SURFACE RECONSTRUCTION OF 3D OBJECTS FROM POINTS CLOUD USING NEURAL NETWORKS: 3D-SOM

Resume :

The work undertaken within the framework of this thesis has for main goal to develop an unsupervised self-organizing neuronal architecture dedicated to surface reconstruction of 3D objects. The entry space of the neural network consists in an unorganized cloud of 3D points acquired on the surface of the 3D object. Our work was directed towards the development of a new neuronal architecture inspired from the self-organizing maps of Kohonen that we call 3D-SOM. After a learning process, the network appears as a neuronal mesh having the shape of the object to reconstruct.

We associated various learning laws to the 3D-SOM, allowing the setting of an effective self-organization process. We also introduced:

- a flexible level of adaptation which allows a dynamic evolution of the learning neighbourhood.
- a local adaptation law subordinated to specific criteria for each neuron of each triangle (local adaptation coefficient, local density of the points cloud, orientations of the normals to the triangles).
- a global and local subdivision process which consists in increasing the number of neurons (thus the number of triangles in the mesh) composing the neural network at the places where the network encounters difficulties of adaptation to the real shape of the object to reconstruct.

Mots clés : reconstruction de surfaces, réseaux de neurones, cartes auto-organisatrices de Kohonen, 3D-SOM, Nuages de points, reconstruction 3D.

Keywords: surface reconstruction, neural networks, Kohonen's self-organizing maps, 3D-SOM, cloud of points, 3D reconstruction

