Pergamon

Neural Networks, Vol. 7, No. 3, pp. 507–522, 1994

0893-6080/94 $6.00 + .00

CONTRIBUTED ARTICLE

Topology Representing Networks

THOMAS MARTINETZ[1,2] AND KLAUS SCHULTEN[1]

[1]University of Illinois and [2]Siemens AG

(*Received* 4 *March* 1993; *accepted* 20 *September* 1993)

Abstract—*A Hebbian adaptation rule with winner-take-all like competition is introduced. It is shown that this competitive Hebbian rule forms so-called Delaunay triangulations, which play an important role in computational geometry for efficiently solving proximity problems. Given a set of neural units i, $i = 1, \ldots, N$, the synaptic weights of which can be interpreted as pointers $\mathbf{w}_i$, $i = 1, \ldots, N$ in $\Re^D$, the competitive Hebbian rule leads to a connectivity structure between the units i that corresponds to the Delaunay triangulation of the set of pointers $\mathbf{w}_i$. Such competitive Hebbian rule develops connections ($C_{ij} > 0$) between neural units i, j with neighboring receptive fields (Voronoi polygons) V_i, V_j, whereas between all other units i, j no connections evolve ($C_{ij} = 0$). Combined with a procedure that distributes the pointers $\mathbf{w}_i$ over a given feature manifold M, for example, a submanifold $M \subset \Re^D$, the competitive Hebbian rule provides a novel approach to the problem of constructing topology preserving feature maps and representing intricately structured manifolds. The competitive Hebbian rule connects only neural units, the receptive fields (Voronoi polygons) V_i, V_j of which are adjacent on the given manifold M. This leads to a connectivity structure that defines a perfectly topology preserving map and forms a discrete, path preserving representation of M, also in cases where M has an intricate topology. This makes this novel approach particularly useful in all applications where neighborhood relations have to be exploited or the shape and topology of submanifolds have to be taken into account.*

Keywords—Proximity problems, Delaunay triangulation, Voronoi polyhedron, Hebb rule, Winner-take-all competition, Topology preserving feature map, Topology representation, Path preservation, Path planning.

1. DELAUNAY TRIANGULATION AND PROXIMITY PROBLEMS

In a number of information processing tasks the problem that has to be solved efficiently can be reduced to a geometric problem that deals with the proximity of points in a metric space. The most prominent example of such a *proximity problem* is the *nearest-neighbor* or, more generally, the *k-nearest-neighbor search:* given N points in a metric space, which is (are) the nearest (k nearest) neighbor(s) to a new given query point (Duda & Hart 1973). This *best match retrieval* has to be performed in classification and interpolation tasks with applications in areas ranging from speech and image processing over robotics to efficient storage and transfer of data (Makhoul, Roucos, & Gish, 1985; Kohonen, Mäkisara, & Saramäki, 1984; Naylor & Li, 1988; Gray, 1984; Nasrabadi & King, 1988; Nasrabadi & Feng, 1988; Ritter & Schulten, 1986). Another example of a proximity problem is the construction of the *Euclidean minimum spanning tree:* given N points in a metric space, what is the graph of minimum total length whose vertices are the given points (Kruskal, 1956; Prim, 1957; Dijkstra, 1959). Constructing the Euclidean minimum spanning tree is a common task in applications requiring optimally designed networks, for example, communication systems that have minimal interconnection length. Other applications of the Euclidean minimum spanning tree are in clustering (Gower & Ross, 1969; Zahn, 1971), pattern recognition (Osteen & Lin, 1974), and in searching for (approximate) solutions of the *traveling salesman problem* (Rosenkrantz, Stearns, & Lewis, 1974). A third example is the *triangulation problem:* given N points in a plane, connect them by nonintersecting straight lines so that every region inside the convex hull of the N points is a triangle. The triangulation problem occurs in the finite-element method (Strang & Fix, 1973) and in function interpolation on the basis of N data points, where the

Acknowledgements: We thank Benoit Noël and Philippe Dalger for pointing out the connection between the competitive Hebb rule and second-order Voronoi polyhedra and for formulating Theorem 1. This work has been supported by the Carver Charitable Trust, the BMFT under Grant No. 01IN102A7, and by a fellowship of the Volkswagen Foundation to T.M.

Requests for reprints should be sent to Klaus Schulten, Beckman-Institute and Department of Physics, University of Illinois at Urbana-Champaign, 405 North Mathews Ave., Urbana, IL 61801.

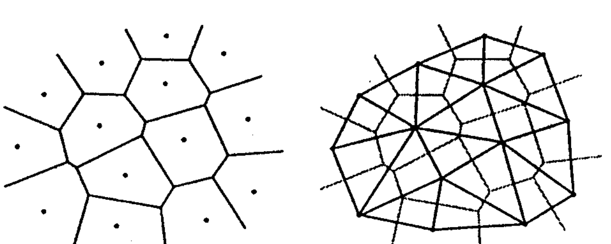
function surface is approximated by a network of triangular facets (George, 1971). A comprehensive overview of the above and further proximity problems can be found in Preparata and Shamos (1985).

A powerful structure from computational geometry that solves or, at least, yields a starting point for efficiently solving the above and other proximity problems is the so-called *Voronoi diagram* and its dual, the *Delaunay triangulation*. The Voronoi diagram $\mathcal{V}_S$ of a set $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ of points $\mathbf{w}_i \in \Re^D$ is given by N D-dimensional polyhedra, the Voronoi polyhedra V_i, which are defined as follows: the Voronoi polyhedron V_i of a point $\mathbf{w}_i \in S$ is given by the set of points $\mathbf{v} \in \Re^D$ that are closer to $\mathbf{w}_i$ than to any other $\mathbf{w}_j \in S$, $j \neq i$, that is,

$$V_i = \{\mathbf{v} \in \Re^D \mid \|\mathbf{v} - \mathbf{w}_i\| \leq \|\mathbf{v} - \mathbf{w}_j\| \, j = 1, \ldots, N\} \qquad i = 1, \ldots, N. \quad (1)$$

The Voronoi polyhedra provide a complete partitioning of the embedding space $\Re^D$, that is, $\Re^D = \cup_{i=1}^{N} V_i$. In the context of *winner-take-all* type neural networks the Voronoi polyhedron V_i is often called the receptive field of neural unit i, with $\mathbf{w}_i$ being the synaptic weight vector of this neural unit. For all input vectors $\mathbf{v} \in V_i$ the element i is the best matching unit of the network and is employed to represent (encode) these input patterns $\mathbf{v}$. Figure 1a illustrates the definition of V_i by showing the Voronoi diagram, that is, all the Voronoi polygons, of a sample set of points in a plane. In $\Re^2$ the Voronoi diagram forms a graph, the vertices of which are given by all the $\mathbf{v} \in \Re^2$ that are simultaneously element of three Voronoi polygons V_i. The edges are given by all the $\mathbf{v} \in \Re^2$ that are simultaneously element of two Voronoi polygons. The Voronoi diagram as the set of *all* Voronoi polyhedra contains all the proximity information about the set of points S.

The straight line dual of the Voronoi diagram is the so-called *Delaunay triangulation* (Delaunay, 1934). In a plane, the Delaunay triangulation is obtained if we connect all pairs $\mathbf{w}_i$, $\mathbf{w}_j \in S$, the Voronoi polygons V_i, V_j of which share an edge. In general, for embedding spaces $\Re^D$ of arbitrary dimension D, the Delaunay triangulation $\mathcal{D}_S$ of a set $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ of points $\mathbf{w}_i \in \Re^D$ is defined by the graph whose vertices are the $\mathbf{w}_i$ and whose *adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$, $i, j = 1, \ldots N$ carries the value one iff $V_i \cap V_j \neq \emptyset$. Two vertices $\mathbf{w}_i$, $\mathbf{w}_j$ are connected by an edge iff their Voronoi polyhedra V_i, V_j are adjacent. An illustration of the planar case is given in Figure 1b, where the Delaunay triangulation that corresponds to the Voronoi diagram of Figure 1a is shown. In a plane, each edge of the Delaunay triangulation corresponds to an edge of the Voronoi diagram, and the two corresponding edges are always perpendicular to each other.



(a) (b)

FIGURE 1. (a) The Voronoi diagram of a set of points. (b) The Delaunay triangulation that corresponds to the Voronoi diagram in (a).

A number of theorems about properties of the Voronoi diagram and the Delaunay triangulation are known (see, e.g., Preparata & Shamos, 1985). However, most of them are valid or at least can be proven only in the planar case, for $D = 2$. In higher-dimensional embedding spaces $\Re^D$, $D > 2$, only little is known so far. One reason is that only for $D = 2$ the Voronoi diagram and the Delaunay triangulation are planar graphs and, therefore, only for $D = 2$ Euler's formula can be applied (Bollobás, 1979). Euler's formula provides the important result that in the planar case the number of edges of the Voronoi diagram as well as of the Delaunay triangulation does not exceed $3N - 6$ and, hence, the Voronoi diagram and the Delaunay triangulation can be stored in only linear space (linear in the number of vertices N). Further, due to this result, both structures are transformable into each other in only linear time.

Constructing the Delaunay triangulation in a preprocessing stage yields a starting point for efficiently solving proximity problems. It can be shown that if the Delaunay triangulation of a given set of points S is known, the above stated and other proximity problems can be solved with at most linearly increasing computational effort. The triangulation problem, for example, is obviously already solved with the construction of the Delaunay triangulation and does not need further computation.[1] The computation time needed for finding the Euclidean minimum spanning tree is reduced significantly because the edges of the Euclidean minimum spanning tree are a subset of the edges of the Delaunay triangulation (Shamos, 1978). Knowing the Delaunay triangulation, it only requires $\mathcal{O}(N)$ instead of $\mathcal{O}(N \log N)$ time for its construction. The nearest-neighbor and k-nearest-neighbor search can be performed in only $\mathcal{O}(\log N)$ instead of $\mathcal{O}(N)$ time by exploiting the Delaunay triangulation (Knuth, 1973).

In the following section we will introduce a Hebbian adaptation rule that, within a network of neural units, yields interneural connections corresponding to the

[1] Solving the triangulation problem by means of the Delaunay triangulation has advantages particularly in function interpolation. When a function is approximated piecewise-linear over the facets of a triangulation, the Delaunay triangulation yields a smaller worst case error than any other triangulation (Omohundro, 1990).

Delaunay triangulation. In Section 3 we will explain the concept of topology preserving feature maps. We will introduce the term *masked Voronoi polyhedron* and show how these masked Voronoi polyhedra provide a rigorous definition for the terms *neighborhood and topology preserving maps*. This leads to *induced Delaunay triangulations* as the graphical structure that provides perfect topology preservation. We will show that the Hebbian adaptation rule of the following section forms induced Delaunay triangulations and, hence, is able to form perfectly topology preserving maps also of topologically intricately structured manifolds, for example, of manifolds that are disconnected and/or piecewise 0-, 1-, 2-, etc., dimensional. In Section 4 we will show that the induced Delaunay triangulations provided by the Hebbian adaptation rule yield path preserving representations and allow one to describe, classify, and plan paths on manifolds. These induced Delaunay triangulations, which reflect the topology and, at the same time, form perfectly topology preserving maps and path preserving representations of given manifolds, we call *topology representing networks* (TRN). In the last section we will present a compact algorithm, derived from the Hebbian adaptation rule of the next section, which constructs TRNs.

2. HEBBIAN ADAPTATION RULE FORMS DELAUNAY TRIANGULATIONS

In the following we assume a set of neural units i, $i = 1, \ldots, N$ that can develop *lateral connections* between each other. A neural unit connects itself with another unit by developing a *synaptic link* to this unit. The lateral connections are described by a connection strength matrix **C** with elements $C_{ij} \in \Re_0^+$. The larger a matrix element C_{ij}, the stronger is the synaptic link from unit i to unit j. Only if $C_{ij} > 0$, we regard neural unit i as being connected with unit j. If $C_{ij} = 0$, neural unit i is *not* connected with unit j. Negative values for C_{ij} do not arise.

The basic principle that governs the change of interneural connection strength, at least in parts of the hippocampus (Kelso, Ganong, & Brown, 1986), was first formulated by Hebb (1949). According to Hebb's postulate, a presynaptic unit i increases the strength of its synaptic link to a postsynaptic unit j if both units are concurrently active, that is, if both activities do correlate. A variety of quantitative formulations of this conjunctive mechanism have been proposed, for example, for modeling Pavlovian conditioning (Grossberg, 1974), motor learning (Marr, 1969), or associative memory (Palm, 1982; Kohonen, Oja, & Lehtio, 1985). In its simplest mathematical formulation Hebb's rule is described by the equation

$$\Delta C_{ij} \propto y_i \cdot y_j, \qquad (2)$$

in which the change of the strength C_{ij} of the synaptic link from unit i to unit j is linearly proportional to the presynaptic activity y_i and to the postsynaptic activity y_j (see, e.g., Cooper, 1973). The quantities y_i, $i = 1, \ldots, N$ denote the output activities of the neural units. More realistic and detailed mathematical descriptions of Hebb's postulate take temporal aspects and the stochastic nature of the output activities of neurons into account. Also, it may be necessary to add a decay term to keep the strength of the synaptic links bounded.

Different neural network models have employed different realizations of Hebb's postulate, depending on the adaptation rules required to achieve the desired information processing task. We will employ the Hebb rule in a form that incorporates the novel aspect of *competition* among the synaptic links. We assume that to each neural unit i a weight vector $\mathbf{w}_i \in \Re^D$ is assigned. Further, we assume that each neural unit i, $i = 1, \ldots, N$ receives the same external input patterns $\mathbf{v} \in \Re^D$. The weight vector $\mathbf{w}_i$ determines the center of the receptive field of unit i in the sense that with the reception of an input pattern $\mathbf{v}$ the output activity y_i of unit i is the larger the closer its $\mathbf{w}_i$ is to $\mathbf{v}$. In mathematical terms, we assume that $y_i = R(\|\mathbf{v} - \mathbf{w}_i\|)$ is valid, with $R(\cdot)$ being a positive and continuously monotonically decreasing function (e.g., a Gaussian).

If we apply the Hebb rule in the simple form given by eqn (2), we obtain for the change of the strength of the lateral connections C_{ij} with the presentation of an input pattern $\mathbf{v}$

$$\Delta C_{ij} \propto R(\|\mathbf{v} - \mathbf{w}_i\|) \cdot R(\|\mathbf{v} - \mathbf{w}_j\|). \qquad (3)$$

Asymptotically, that is, with the sequential presentation of many input patterns $\mathbf{v} \in \Re^D$, the C_{ij}'s are determined by the integral of eqn (3) over the given pattern distribution $P(\mathbf{v})$. If we assume a pattern distribution that is homogeneous over $\Re^D$, we obtain

$$\Delta C_{ij}(t \to \infty) \propto \int_{\Re^D} R(\|\mathbf{v} - \mathbf{w}_i\|) \cdot R(\|\mathbf{v} - \mathbf{w}_j\|)\, d\mathbf{v}. \qquad (4)$$

Hence, employing the Hebb rule in the simple form as given in eqn (2) yields the rather trivial result that each neural unit i develops connections to all the other units $j \neq i$, with lateral connection strengths C_{ij} that are simply proportional to the overlaps of the receptive fields $R(\|\mathbf{v} - \mathbf{w}_i\|)$ and $R(\|\mathbf{v} - \mathbf{w}_j\|)$. The strength of the synaptic link between two units i and j is simply monotonically and continuously decreasing with the distance between $\mathbf{w}_i$ and $\mathbf{w}_j$.

As in many systems governed by self-organizing processes, the connectivity pattern that evolves on the set of neural units becomes significantly more structured if we introduce competition. In a winner-take-all network, for example, the units compete with each other based on their output activities, which finally leads to an adaptation only of the weights of the unit with the highest output activity (see, e.g., Grossberg, 1976).

Without competition, all the units would behave alike and no specialization of the units, as it is characteristic for winner-take-all networks, would evolve.

Analog to the competition among the units in a winner-take-all network we introduce competition among the synaptic links. Instead of being based on the output activities of the neural units itself, as in a winner-take-all network, in our model the competition among the synaptic links is determined by what we want to call the *correlated* output activities Y_{ij}, the correlations of the output activities of all pairs of pre- and postsynaptic units. In the quantitative formulation given below, the correlated output activities are determined by $Y_{ij} = y_i \cdot y_j$, according to the Hebb rule (2). Keeping the analogy to winner-take-all networks, with the presentation of an input pattern $\mathbf{v}$ we only modify the synaptic link $i - j$ whose activity $Y_{ij} = y_i \cdot y_j$ is highest. Instead of changing the connection strengths C_{ij} according to the Hebb rule (2), in the following we will employ a winner-take-all or competitive version of eqn (2), determined by

$$\Delta C_{ij} \propto \begin{cases} y_i \cdot y_j & \text{if} \quad y_i \cdot y_j \geq y_k \cdot y_l \quad \forall k, l = 1, \ldots, N \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

We will show that, instead of connecting each unit with all the other units, the competitive Hebb rule (5) forms a connectivity structure among the neural units $i, i = 1, \ldots, N$ that corresponds to the Delaunay triangulation of the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_N$. More precisely, we will show that if we present sequentially input patterns $\mathbf{v}$ with a distribution $P(\mathbf{v})$ that has support (is nonzero) everywhere on $\Re^D$, then the elements C_{ij} of the connection strength matrix $\mathbf{C}$ obey asymptotically

$$\theta[C_{ij}(t \rightarrow \infty)] = A_{ij} \quad i, j = 1, \ldots, N \tag{6}$$

where $\theta(\cdot)$ is the Heavyside step function and where A_{ij} are the elements of the adjacency matrix $\mathbf{A}$ of the Delaunay triangulation of the points $\mathbf{w}_1, \ldots, \mathbf{w}_N$, for which

$$A_{ij} = \begin{cases} 1 & \text{if } V_i \cap V_j \neq \emptyset \quad (V_i, V_j \text{ are adjacent}) \\ 0 & \text{if } V_i \cap V_j = \emptyset \quad (V_i, V_j \text{ are not adjacent}) \end{cases} \tag{7}$$

is valid. V_i, V_j again denote the Voronoi polyhedra of $\mathbf{w}_i, \mathbf{w}_j$, that is, in a winner-take-all network the receptive fields of the units i, j.

Before we prove eqn (6), we first show how the competitive Hebb rule for constructing the Delaunay triangulation can be formulated in a very compact algorithm. The correlated activity $Y_{ij} = y_i \cdot y_j$ is highest if i (or j) denotes the neural unit with the largest output and j (or i, respectively) denotes the unit with the second largest output. Because $Y_{ij} = Y_{ji}$ is valid, there are always two *winning* links, the strengths of which are changed by the same amount, that is, the strength of the link from i to j as well as of the link from j to i. Hence, the connection strength matrix C_{ij} is symmetric, and if we say "unit i is connected with unit j," this always implies that unit j is also connected with unit i. Because we are interested in the graph defined by the connections between the units, and because this graph is completely determined by the adjacency matrix $A_{ij} = \theta(C_{ij})$, in the following algorithm we will not care about the absolute values of the C_{ij}'s, but will only register whether a C_{ij} becomes nonzero.

The procedure for constructing the connections between the units i, j ($i, j = 1, \ldots, N$) can now be formulated as follows:

(i) Set all connection strengths C_{ij} to zero;

(ii) Present an input pattern $\mathbf{v} \in \Re^D$ with probability $P(\mathbf{v})$;

(iii) Determine unit i_0 for which

$$\|\mathbf{v} - \mathbf{w}_{i_0}\| \leq \|\mathbf{v} - \mathbf{w}_j\| \quad \forall j = 1, \ldots, N$$

and unit i_1 for which

$$\|\mathbf{v} - \mathbf{w}_{i_1}\| \leq \|\mathbf{v} - \mathbf{w}_j\| \quad \forall j \neq i_0, \quad j = 1, \ldots, N;$$

(iv) If $C_{i_0 i_1} = 0$, set $C_{i_0 i_1} > 0$, that is, connect i_0 and i_1. If $C_{i_0 i_1} > 0$, leave $C_{i_0 i_1}$, unchanged. Continue with (ii).

In Figure 2 algorithm (i)–(iv) is illustrated. Two units i, j become connected iff their Voronoi polyhedra V_i, V_j are adjacent. To show rigorously that the adjacency matrix $A_{ij} = \theta(C_{ij})$ of the connectivity structure becomes equivalent to the adjacency matrix of the Delaunay triangulation $\mathcal{D}_S$ of the set of points $S = (\mathbf{w}_1,$

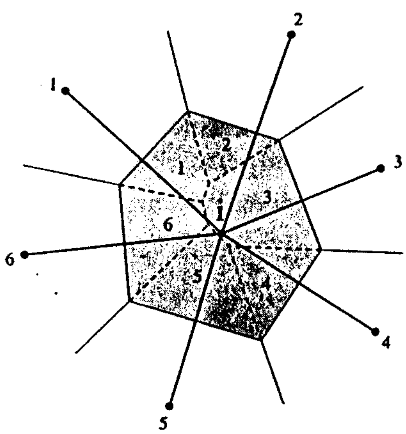


FIGURE 2. Illustration of the update rule for the connections between the neural units. Each time an input pattern v is presented within the shaded area, which depicts the Voronoi polygon V_i of unit i, a connection from i to the unit j with its weight vector $\mathbf{w}_j$ second closest to the input pattern is established. The numbers 1, . . . , 6 denote the subregions within the shaded area for which the respective neighboring unit is second closest to an input signal $\mathbf{v} \in V_i$. Unit i develops connections only to those units, the Voronoi polygons of which share an edge with its own Voronoi polygon.

$\ldots, \mathbf{w}_N$), we introduce the *second-order Voronoi polyhedra* V_{ij}, $i, j = 1, \ldots, N$. The second-order Voronoi polyhedron V_{ij} is given by all the $\mathbf{v} \in \Re^D$ for which $\mathbf{w}_i$ and $\mathbf{w}_j$ are the two closest points of S; that is, V_{ij} is defined by

$$V_{ij} = \{\mathbf{v} \in \Re^D \mid \|\mathbf{v} - \mathbf{w}_i\| \le \|\mathbf{v} - \mathbf{w}_k\| \wedge \|\mathbf{v} - \mathbf{w}_j\| \le \|\mathbf{v} - \mathbf{w}_k\| \; \forall k \ne i, j\}. \quad (8)$$

As V_i, also V_{ij} forms a convex polyhedron. We see from eqn (8) that algorithm (i)–(iv) connects two units i, j only if $V_{ij} \ne \emptyset$ is valid. We will prove that $V_{ij} \ne \emptyset$ is valid iff the corresponding first-order Voronoi polyhedra V_i, V_j are adjacent, that is, iff $V_i \cap V_j \ne \emptyset$. Then, in case $\int_{V_{ij}} P(\mathbf{v})\, d\mathbf{v} \ne 0$ holds for each $V_{ij} \ne \emptyset$, the connections generated by algorithm (i)–(iv) form the Delaunay triangulation of the points $\mathbf{w}_1, \ldots, \mathbf{w}_N$.[2]

THEOREM 1. *For a set* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *of points* $\mathbf{w}_i \in \Re^D$ *the relation*

$$V_i \cap V_j \ne \emptyset \Leftrightarrow V_{ij} \ne \emptyset \quad (9)$$

is valid for each pair i, j. V_i *denotes the first-order Voronoi polyhedron of point* $\mathbf{w}_i$, *and* V_{ij} *denotes the second-order Voronoi polyhedron of the points* $\mathbf{w}_i$, $\mathbf{w}_j$.

Proof. If $V_i \cap V_j \ne \emptyset$ is valid, there is a $\mathbf{v} \in \Re^D$ with $\mathbf{v} \in V_i$ and $\mathbf{v} \in V_j$. Then we obtain $\|\mathbf{v} - \mathbf{w}_i\| = \|\mathbf{v} - \mathbf{w}_j\| \le \|\mathbf{v} - \mathbf{w}_k\|$ for all $\mathbf{w}_k \in S$ and, therefore, $\mathbf{v} \in V_{ij}$, that is, $V_{ij} \ne \emptyset$, is valid. ■

If $V_{ij} \ne \emptyset$ is valid, there is a $\mathbf{v} \in \Re^D$ for which the points $\mathbf{w}_i$ and $\mathbf{w}_j$ are the two nearest neighbors. Without loss of generality we assume that $\mathbf{w}_i$ is the nearest neighbor. Because for each $\mathbf{u} \in \overline{\mathbf{v}\mathbf{w}_j}$ the point $\mathbf{w}_j$ is either the nearest or the second nearest neighbor of $\mathbf{u}$, and for $\mathbf{u} = \mathbf{v}$ the point $\mathbf{w}_i$ is closest and for $\mathbf{u} = \mathbf{w}_j$ the point $\mathbf{w}_j$ is closest to $\mathbf{u}$, there is a $\mathbf{u}^* \in \overline{\mathbf{v}\mathbf{w}_j}$ for which $\|\mathbf{u}^* - \mathbf{w}_i\| = \|\mathbf{u}^* - \mathbf{w}_j\|$ is valid. Hence, we obtain $\mathbf{u}^* \in V_i$ and $\mathbf{u}^* \in V_j$, and, therefore, $\mathbf{u}^* \in V_i \cap V_j$, that is, $V_j \cap V_j \ne \emptyset$, is valid.

In the following we will consider pattern distributions $P(\mathbf{v})$ that have support not on the entire embedding space $\Re^D$, but only on a submanifold M. In these cases, for some $V_{ij} \ne \emptyset$ the integral $\int_{V_{ij}} P(\mathbf{v})\, d\mathbf{v}$ might vanish, with the result that the edge $i - j$ will not be established by the competitive Hebb rule. In these cases, the competitive Hebb rule does not form the entire Delaunay triangulation, but only subgraphs of it. We will show that these subgraphs define topology preserving maps.

3. DELAUNAY TRIANGULATION AND TOPOLOGY PRESERVING MAPS

The definition of the Delaunay triangulation as being the graph that connects those points $\mathbf{w}_i$, $\mathbf{w}_j$ that have adjacent Voronoi polyhedra V_i, V_j makes this structure ideally suited for a proximity problem of a different type than the ones mentioned in the first section, namely, for the formation of so-called topology preserving feature maps. Topology preserving feature maps play an important role as components in a variety of natural as well as artificial neural information processing systems (Knudsen, de Lac, & Esterly, 1987; Kohonen, 1989; Ritter, Martinetz, & Schulten, 1992). By projecting input patterns onto a network of neural units such that similar patterns are projected onto units adjacent in the network and, vice versa, such that units adjacent in the network code similar patterns, a representation of the input patterns is achieved that in postprocessing stages allows one to exploit the similarity relations of the input patterns. Examples of topology preserving feature maps in the nervous system are the retinotopic map in the visual cortex (Hubel & Wiesel, 1974; Blasdel & Salama, 1986), the mapping from the body surface onto the somatosensory cortex (Kaas et al., 1979), or the tonotopic maps in the auditory cortex (Suga & O'Neill, 1979). As components of artificial neural information processing systems, topology preserving feature maps have been applied successfully in speech processing (Kohonen et al., 1984; Kohonen, 1990; Naylor & Li, 1988; Brandt, Behme, & Strube, 1991), image processing (Nasrabadi & Feng, 1988), and robotics (Ritter & Schulten, 1986; Martinetz, Ritter, & Schulten, 1990).

A topology preserving feature map is determined by a mapping Φ from a manifold $M \subseteq \Re^D$ onto the vertices (neural units) i, $i = 1, \ldots, N$ of a graph (network) G. The mapping from M to G is determined by pointers $\mathbf{w}_i \in \Re^D$, $i = 1, \ldots, N$ attached to the vertices i. A feature vector $\mathbf{v} \in M$ is mapped to the vertex $i^*(\mathbf{v})$, the pointer $\mathbf{w}_{i^*(\mathbf{v})}$ of which is closest to $\mathbf{v}$; that is, $\mathbf{v}$ is mapped to the vertex $i^*(\mathbf{v})$ whose Voronoi polyhedron $V_{i^*(\mathbf{v})}$ encloses $\mathbf{v}$. The mapping Φ is completely determined by the pointer set $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$, and we can write

$$\Phi_S: M \mapsto G, \quad \mathbf{v} \in M \mapsto i^*(\mathbf{v}) \in G \quad (10)$$

where $i^*(\mathbf{v})$ is determined by

$$\|\mathbf{w}_{i^*(\mathbf{v})} - \mathbf{v}\| \le \|\mathbf{w}_i - \mathbf{v}\| \quad \forall i \in G. \quad (11)$$

The mapping Φ_S from M to G is neighborhood preserving, if similar feature vectors, that is, $\mathbf{v}$'s that are close within the manifold M, are mapped to vertices i that are close within the graph. This requires that pointers $\mathbf{w}_i$, $\mathbf{w}_j$ that are neighboring on the feature manifold are assigned to vertices i, j that are adjacent in the graph.[3]

The inverse mapping Φ_S^{-1} from the graph G onto the manifold M is determined by

$$\Phi_S^{-1}: G \mapsto M, \quad i \in G \mapsto \mathbf{w}_i \in M \quad (12)$$

[2] The following theorem was formulated together with Philippe Dalger and Bennoit Noël (Dalger et al., 1992).

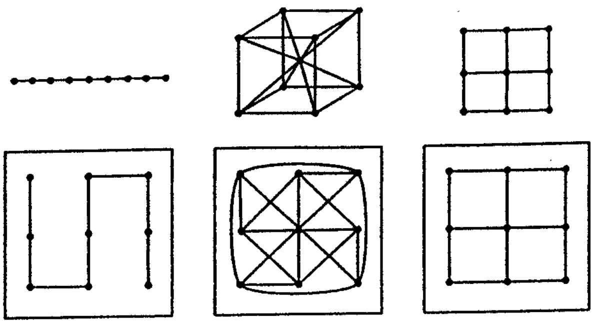


FIGURE 3. A square-shaped manifold M mapped onto three different graphs G. Each of the graphs consists of nine vertices i, and the associated pointers w_i are distributed regularly over the square. The three different feature maps are visualized by marking the pointer positions w_i on the manifold M by dots and connecting by lines those pointer positions w_i, w_j the corresponding vertices i, j of which are adjacent in the graph G. In (a) the graph is a string that only allows a neighborhood preserving inverse mapping from G to M, but not a neighborhood preserving mapping from M to G. Adjacent vertices are assigned to neighboring locations, but not all the pointers w_i that are neighboring on M belong to adjacent vertices. With this graphical structure it is not possible to form a topology preserving map of M. In (b) the graph G has a three-dimensional structure. In this case the mapping from M to G, but not the inverse mapping from G to M, is neighborhood preserving. Pointers that are neighboring on M are assigned to adjacent vertices, but not all the vertices that are adjacent in G are assigned to neighboring locations on M. Hence, as in (a), the given graphical structure does not allow one to form a topology preserving map of M. Only in (c) where the graph G is a square lattice and matches the topology of M, both the mapping from M to G and the inverse mapping from G to M are neighborhood preserving. Only in this case does G form a topology preserving map of M.

(we assume that all pointers w_i, $i = 1, \ldots, N$, lie on M). Analog to neighborhood preservation of Φ_S, neighborhood preservation of the inverse mapping Φ_S^{-1} is given if pointers w_i, w_j of adjacent vertices i, j are neighboring on the manifold M.

The graph G with its vertices i assigned to the locations w_i forms a topology preserving map of M, if the mapping Φ_S from M to G as well as the inverse mapping Φ_S^{-1} from G to M is neighborhood preserving. Only then G forms a map on which adjacent locations (vertices) correspond to features neighboring on M and, vice versa, which represents features neighboring on M by adjacent locations.

The question arises which type of graph G can provide a topology preserving map of a given manifold M. Crucial is G's connectivity structure or topology compared to the topology of M. This is demonstrated schematically in Figure 3. In each of the three cases that are depicted the manifold M is simply a square, and the graph G consists of nine vertices i, the associated pointers w_i of which are distributed regularly over the square-shaped manifold. In Figure 3a the graph G has a one-dimensional topology and consists of nine vertices arranged as a string, compared to the two-dimensional manifold M. In this case G cannot form a topology preserving map of M. All it can achieve is a neighborhood preserving inverse mapping Φ^{-1} from G to M. In Figure 3b we have the opposite case. The dimensionality of G is higher than the dimensionality of M. The result is that a neighborhood preserving mapping Φ from M to G is possible, but not a neighborhood preserving inverse mapping Φ^{-1} from G to M. Hence, as in (a), G is not able to form a topology preserving map of M. In Figure 3c the topology of the graph G is a square lattice and corresponds to the topology of the manifold M. Only in this case does G form a topology preserving map of M; that is, the mapping Φ from M to G as well as the inverse mapping Φ^{-1} from G to M is neighborhood preserving.

A number of neural network models for adaptively forming topology preserving feature maps have been proposed (Willshaw & von der Malsburg, 1976; von der Malsburg & Willshaw, 1977; Takeuchi & Amari, 1979; Kohonen, 1982a, b; Durbin & Mitchison, 1990). A model that provides a very compact procedure and, therefore, has found widespread application in artificial neural information processing systems is Kohonen's self-organizing feature map (Kohonen, 1982a, b, 1989, 1990). This algorithm requires that one first chooses a graph G, usually a one-, two-, or three-dimensional lattice. In a subsequent adaptation procedure, the pointers w_i are distributed over the given feature manifold M in such a way, that (i) pointers lie only on M, and (ii) pointers of vertices adjacent in G are assigned to locations close on M. Hence, Kohonen's algorithm tries to form a neighborhood preserving inverse mapping Φ^{-1} from G to M, but not necessarily a neighborhood preserving mapping Φ from M to G. To obtain a topology preserving map, the topological structure of the preset graph G has to match the topological structure of the given manifold M.

Figure 4 illustrates three examples of different maps that evolve with Kohonen's algorithm, depending on the degree of mismatch between the topological structure of the preset graph G and the given manifold M. In all three examples, the manifold M is a submanifold of $\Re^2$. In Figures 4a, b, the manifold M again is a square, like in Figure 3. In Figure 4c the manifold M is disconnected and piecewise one- and two-dimensional. In Figure 4a the preset graph G is a two-dimensional lattice and, hence, the topology of the given manifold and the graph do match. As we can see, in this case Kohonen's self-organizing feature map algorithm provides a topology preserving map. In Figure 4b the topology of

[3] At this point we leave the definition of "adjacency of pointers w_i on a manifold M" to the reader's intuition, like in most contributions on topology preserving feature maps. A rigorous definition based on Voronoi polyhedra is given in the next section.

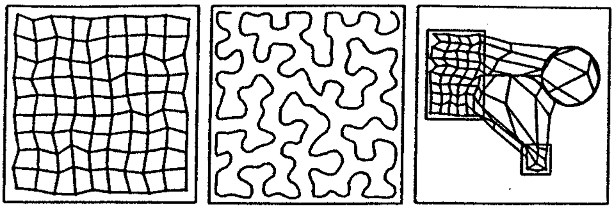


FIGURE 4. Maps of manifolds formed by Kohonen's self-organizing feature map algorithm. In (a) the graph G, a square lattice, has a topology that corresponds to the topology of the square-shaped feature manifold *M*. Hence, the map that evolves is topology preserving. In (b) the given manifold *M* again is a square; however, this time the graph is a string and does not correspond topologically to *M*. With this graph it is not possible to achieve a topology preserving map, and the Kohonen algorithm can only form a neighborhood preserving inverse mapping Φ^{-1} from G to *M*, but not a neighborhood preserving mapping Φ from *M* to G. Part (c) shows an example of very high topological mismatch between *M* and G. The given manifold *M* is disconnected and consists of a two-dimensional (a rectangle and a square) and a one-dimensional (a line and a circle) submanifold; the graph G is a square-lattice. Again no topology preservation is possible. In this case the Kohonen algorithm neither achieves a neighborhood preserving mapping Φ from *M* to G nor a neighborhood preserving inverse mapping Φ^{-1} from G to *M*.

the given manifold and of the graph do not match. The graph is a one-dimensional string of vertices, in contrast to the two-dimensional feature manifold. As shown schematically in Figure 3a, because of this mismatch topology preservation is not possible, and Kohonen's algorithm can only provide a neighborhood preserving inverse mapping Φ^{-1} from G to M. Figure 4c finally shows an example where the topological structure of the graph G and the given manifold M mismatch to an extent that neither a neighborhood preserving inverse mapping Φ^{-1} from G to M nor a neighborhood preserving mapping Φ from M to G can be achieved by Kohonen's algorithm. The graph G is again a two-dimensional lattice. The given manifold M, however, has a relatively intricate topological structure. It consists of a combination of two-dimensional and one-dimensional partially disconnected submanifolds. The two-dimensional submanifold is formed by a rectangle and a square. The one-dimensional submanifold consists of a circle and a connecting line between the rectangle and the circle. Figure 4c shows that in this case some vertices that are adjacent in G are assigned to locations remote on the manifold M, and, vice versa, feature vectors adjacent on M may be mapped onto vertices remote in G. Obviously, the topology preservation of the map is highly disturbed.

In Figure 3 we have demonstrated schematically that topology preserving maps are only possible if the topological structure of the graph G matches the topological structure of the given manifold M. The Kohonen algorithm starts from a prespecified graph G. The three examples of Figure 4 demonstrate that Kohonen's algorithm tries to form a neighborhood preserving inverse mapping from G to M and indeed succeeds in constructing a topology preserving map, if the topological structure of the preset graph G matches the topological structure of the manifold M. In cases, however, where it is not possible to *a priori* determine an appropriate graph G, for example, in cases where the topological structure of M is not known *a priori* or is too complicated to be specified, Kohonen's algorithm necessarily fails in providing perfectly topology preserving maps.

An application where the topological structure of M is not known and, at the same time, a perfectly topology preserving map of M is essential for optimal performance, is in speech recognition as described in Brandt et al. (1991). In their word recognition scheme each word is described by a sequence of 19-dimensional feature vectors, with each feature vector describing the frequency spectrum at a different time step. Hence, a word is a trajectory in a 19-dimensional input space, given typically by a sequence of 40–50 feature vectors. The feature vectors that occur with the spoken words form a lower-dimensional submanifold M within the 19-dimensional input space. By constructing a topology preserving map of the manifold M, as it was first described by Kohonen et al. (1984), each word can be represented by a trajectory within the graph G, that is, by a sequence of *active* nodes. Interestingly, it turns out that a classification of the words based on their trajectories in G yields a significantly increased performance compared to a classification based on their trajectories on M. To obtain optimal performance in this word recognition scheme, the topological structure of G has to match the topology of the feature manifold M. This manifold, however, has a topological structure that is complicated and not known. Its average dimension, for example, seems to lie between two and three (Bauer & Pawelzik, 1992).

Another application where only a neighborhood preserving mapping from M to G together with a neighborhood preserving inverse mapping from G to M, that is, a perfectly topology preserving map of M, provides optimal results is in representing and learning input–output relations with topology preserving maps, for example, in robotics (Ritter & Schulten, 1986; Martinetz et al., 1990). For representing input–output relations, to each vertex i of G an output quantity $\mathbf{a}_i$ is assigned. The $\mathbf{a}_i$'s determine the output of the map for a given input vector $\mathbf{v}$. It can be shown that a robust and fast learning procedure for the $\mathbf{a}_i$'s can only be obtained if the graph G preserves the neighborhood relations of the input manifold, which then allows a concerted adaptation of the output quantities of adjacent vertices (Ritter et al., 1992). Only if adjacent vertices are assigned to locations neighboring on the input manifold, the learning process will be successful. The learning procedure is optimized if, also, all the pointers

w_i that are neighboring on the input manifold correspond to adjacent vertices. This is valid iff the map of the input manifold is perfectly topology preserving. In many applications, however, the input manifold is a submanifold of a high-dimensional input space and may neither be known *a priory* nor topologically simple enough for prespecifying a correspondingly structured graph.

For the applications described above it would be highly desirable to have a procedure that adapts the topology of the graph G to the topology of the given manifold M or, at least, to have a means by which one can decide whether a chosen graph is appropriate for forming a topology preserving map of a given manifold. An approach of the first kind has been proposed by Kohonen and coworkers (Kangas, Kohonen, & Laaksonen, 1990). The idea of this approach is (i) to determine the minimum spanning tree between the points w_i at different stages of their adaptation process and (ii) to take the resulting minimum spanning tree as the graph G that is appropriate at the respective stage. Because the w_i's at the end of the adaptation procedure are distributed only over the manifold M, the final minimum spanning tree will reflect at least some of the topological properties of M. The definition of the minimum spanning tree, however, only requires adjacent vertices i to be assigned to neighboring locations w_i, not vice versa. Pointers w_i that are adjacent on M might very well belong to remote vertices. Therefore, except for special cases, this approach is not able to provide a perfectly topology preserving map. In general, only the inverse mapping Φ^{-1} from G to M but not the mapping Φ from M to G will be neighborhood preserving.

Another approach of the first kind has been introduced by Fritzke (1991). His approach employs two-dimensional, triangular cell structures that are distributed over the manifold M. By selectively adding and removing cells, based on a heuristic criterion that takes a cell's average description error into account, a graph is formed that resembles the shape of the manifold M.

An approach of the second kind for obtaining an appropriate graph G has been proposed by Bauer and Pawelzik (1992). They employ the so-called *topographic product* as a means to determine the degree to which a particular graphical structure is able to preserve the neighborhood relations of a given manifold. By testing, for example, one-, two-, or three-dimensional lattices, one can finally choose the one that preserves the neighborhood relations best.

In the next section we will introduce a novel approach to the problem of constructing topology preserving maps, an approach of the first kind based on the Delaunay triangulation. Because the Delaunay triangulation connects those vertices i the assigned pointers w_i of which are adjacent by having neighboring Voronoi polyhedra V_i, this structure is ideally suited for this purpose. The approach taken is opposite to that of Kohonen's self-organizing feature map. Not the graph G but the pointers w_i are prespecified, for example, by using a vector quantization procedure that distributes them over M. Subsequently, the appropriate edges between the w_i's are constructed for forming a graph G that defines a perfectly topology preserving map of the given manifold M.

3.1. Definition of Topology Preserving Maps

In the previous section the definition of neighborhood preservation of a mapping Φ from M to G or of an inverse mapping Φ^{-1} from G to M was, like in most contributions on topology preserving feature maps, not rigorous. The problem is that adjacency of vertices i in a graph G is clearly defined, a proper definition for adjacency of pointers w_i on M, however, is not obvious. This is why the definition of neighborhood preservation of feature maps has usually been left to the reader's intuition.

An exception is the trivial one-dimensional case. Obviously, two points w_i, w_j in $\Re$ are neighboring if there is no point w_k in between. Expressed in terms of Voronoi polyhedra, an equivalent definition is: two points w_i, w_j in $\Re$ are neighboring if their Voronoi polyhedra are adjacent, that is, if $V_i \cap V_j \neq \emptyset$. In these terms a generalization to higher-dimensional embedding spaces $\Re^D$ is straightforward. However, because we are interested in a definition of neighborhood of points *on a manifold M*, we introduce the *masked Voronoi polyhedron*. The masked Voronoi polyhedron $V_i^{(M)}$ is the part of V_i that is also part of M, that is, $V_i^{(M)} = V_i \cap M$. The superscript indicates the dependence of the masked Voronoi polyhedron on the given manifold M. By using the neighborhood of the masked Voronoi polyhedra $V_i^{(M)}$, $V_j^{(M)}$ instead of the neighborhood of the Voronoi polyhedra V_i, V_j for determining neighborhood of points w_i, w_j on M, we ensure that two points w_i, w_j are called *adjacent on M* only if they do not belong to disconnected regions of M. This leads to the following definition:

DEFINITION 1. *Let* $M \subseteq \Re^D$ *be a given manifold and* $S = \{w_1, \ldots, w_N\}$ *be a set of points* $w_i \in M$. *The Voronoi polyhedra of S are denoted by* V_i, $i = 1, \ldots, N$. *Two points* w_i, $w_j \in M \subseteq \Re^D$ *are adjacent on M if their masked Voronoi polyhedra* $V_i^{(M)} = V_i \cap M$, $V_j^{(M)} = V_j \cap M$ *are adjacent, that is, if* $V_i^{(M)}$ *and* $V_j^{(M)}$ *share an element* $v \in M$ *or, equivalently, if* $V_i^{(M)} \cap V_j^{(M)} \neq \emptyset$ *is valid.*

Each masked Voronoi polyhedron is a part of the manifold M, and, instead of forming a complete partitioning of the embedding space $\Re^D$ like the Voronoi polyhedra, the masked Voronoi polyhedra form a complete partitioning only of the manifold M, that is, $M = \bigcup_{i=1}^{N} V_i^{(M)}$ is valid.

Analog to the definition of the Delaunay triangulation $\mathcal{D}_S$, which is based on the Voronoi polyhedra, we define the *induced Delaunay triangulation* $\mathcal{D}_S^{(M)}$ through the masked Voronoi polyhedra:

DEFINITION 2. *Let* $M \subseteq \Re^D$ *be a given manifold and* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of points* $\mathbf{w}_i \in M$. *The induced Delaunay triangulation* $\mathcal{D}_S^{(M)}$ *of S, given M, is determined by the graph that connects two points* $\mathbf{w}_i$, $\mathbf{w}_j$ *iff their masked Voronoi polyhedra* $V_i^{(M)}$, $V_j^{(M)}$ *are adjacent, that is, by the graph whose adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$, $i, j = 1, \ldots, N$ *has the property*

$$A_{ij} = 1 \Leftrightarrow V_i^{(M)} \cap V_j^{(M)} \neq \emptyset. \qquad (13)$$

It is clear from the definition that the induced Delaunay triangulation forms a subgraph of the Delaunay triangulation.

With these definitions the terms neighborhood preserving mapping, neighborhood preserving inverse mapping, and topology preserving map can be formulated rigorously:

DEFINITION 3. *Let G be a graph (network) with vertices (neural units) i,* $i = 1, \ldots, N$ *and edges (connections) defined by an adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$. *Let* $M \subseteq \Re^D$ *be a given manifold of a D-dimensional embedding space and* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of pointers* $\mathbf{w}_i \in M$ *each of which is attached to a vertex i of the graph G. A mapping* Φ_S *from M to G, defined by*

$$\Phi_S: M \mapsto G, \quad \mathbf{v} \in M \mapsto i^*(\mathbf{v}) \in G \qquad (14)$$

with $i^*(\mathbf{v})$ *as the vertex for which* $\mathbf{v} \in V_{i^*(\mathbf{v})}^{(M)}$ *is valid, is neighborhood preserving, if pointers* $\mathbf{w}_i$, $\mathbf{w}_j$ *that are adjacent on M are assigned to vertices i, j that are adjacent in G.*

DEFINITION 4. *Let G be a graph with vertices i,* $i = 1, \ldots, N$ *and edges defined by an adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$. *Let* $M \subseteq \Re^D$ *be a given manifold of a D-dimensional embedding space and* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of pointers* $\mathbf{w}_i \in M$ *each of which is attached to a vertex i of the graph G. The inverse mapping* Φ_S^{-1} *from G to M, defined by*

$$\Phi_S^{-1}: G \mapsto M, \quad i \in G \mapsto \mathbf{w}_i \in M, \qquad (15)$$

is neighborhood preserving, if vertices i, j that are adjacent in G are assigned to locations $\mathbf{w}_i$, $\mathbf{w}_j$ *that are neighboring on M.*

DEFINITION 5. *Let G be a graph with vertices i,* $i = 1, \ldots, N$ *and edges defined by the adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$. *Let* $M \subseteq \Re^D$ *be a given manifold of a D-dimensional embedding space and* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of pointers* $\mathbf{w}_i \in M$ *each of which is attached to a vertex i of the graph G. The graph G with its vertices i assigned to the locations* $\mathbf{w}_i \in M$ *forms a perfectly topology preserving map, iff the mapping* Φ_S *from M to G as well as the inverse mapping* Φ_S^{-1} *from G to M is neighborhood preserving.*

In Figure 5 the above definitions are illustrated. In Figure 5a–d the given manifold M, which is disconnected, is depicted by the two shaded areas. In Figure 5a the graph G is the Delaunay triangulation, which connects points $\mathbf{w}_i$ that have adjacent Voronoi polygons V_i. According to the above definitions the mapping from M to G is neighborhood preserving, but the inverse mapping from G to M we do not consider as being neighborhood preserving. Some vertices that are connected in the graph are assigned to points that belong to disconnected regions of M, that is, are assigned to masked Voronoi polygons that are not adjacent. In Figure 5b, G is a subgraph of the Delaunay triangulation in (a) and defines a neighborhood preserving inverse mapping from G to M but not a neighborhood preserving mapping from M to G. Vertices connected in G are assigned to masked Voronoi polygons that are adjacent; however, masked Voronoi polygons that are neighboring are not necessarily assigned to adjacent vertices. In Figure 5c the graph G is the minimum spanning tree of the set of points $\mathbf{w}_i$, as it would be suggested by the approach of Kangas et al. (1990). The minimum spanning tree also forms a subgraph of the Delaunay triangulation in (a). With the minimum spanning tree neither the

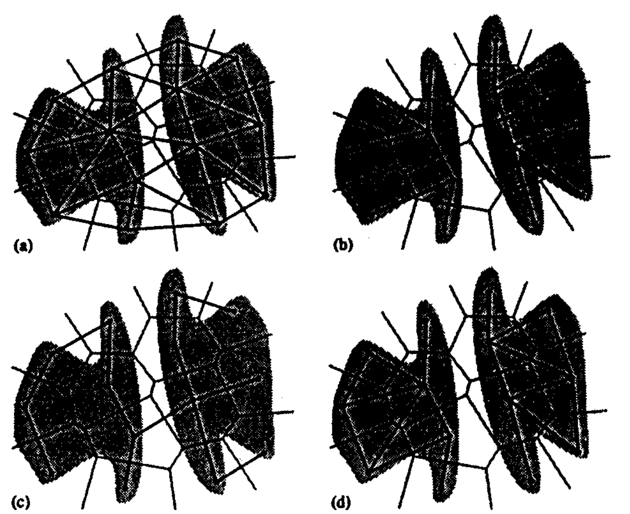


FIGURE 5. Illustration of our definition of neighborhood preserving mappings. In the four examples the given manifold *M* is disconnected and depicted by the two shaded areas. In (a) the graph G is the Delaunay triangulation of the pointers $\mathbf{w}_i$ (see Figure 1). The resulting mapping from *M* to G is neighborhood preserving; however, the mapping from G to *M* we do not consider as being neighborhood preserving because some neighboring vertices belong to masked Voronoi polyhedra that are not adjacent. In (b) the graph G is a subgraph of the Delaunay triangulation and defines a neighborhood preserving inverse mapping from G to *M*, but not a neighborhood preserving mapping from *M* to G. In (c) the graph G is the minimum spanning tree for which neither the mapping from *M* to G nor the inverse mapping from G to *M* is neighborhood preserving. In (d) the graph G is the induced Delaunay triangulation. The mapping from *M* to G as well as the mapping from G to *M* is neighborhood preserving. Pointers and only pointers the Voronoi polygons of which are adjacent *on M*, that is, pointers the *masked* Voronoi polygons of which are adjacent, are connected. In this case the graph G forms a perfectly topology preserving map that reflects the topological structure of the given manifold *M*.

mapping from M to G nor the inverse mapping from G to M is neighborhood preserving. In Figure 5d the graph G is the induced Delaunay triangulation of the set of points $\mathbf{w}_i$ and is again a subgraph of the Delaunay triangulation in (a). With the induced Delaunay triangulation the mapping from M to G as well as the inverse mapping from G to M is neighborhood preserving. Two vertices are connected by an edge iff their masked Voronoi polygons are adjacent. Only in Figure 5d does the graph G form a perfectly topology preserving map of the given manifold M.

It is not surprising that the Delaunay triangulation and the minimum spanning tree in general do not define a perfectly topology preserving map of M. Both graphs are determined only by the set of points $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$. The topology of the manifold M influences both graphical structures only very indirectly. Only through the condition that each $\mathbf{w}_i$ lies on M does the given manifold shape the Delaunay triangulation or the minimum spanning tree. This is different for the induced Delaunay triangulation. The induced Delaunay triangulation is defined through the adjacency of the masked Voronoi polyhedra $V_i^{(M)} = V_i \cap M$. Because each masked Voronoi polyhedron is completely a part of M, the adjacency of these parts carries immediate information about the topological structure of the manifold M.

The above considerations and definitions lead to the following theorem:

THEOREM 2. *Let G be a graph (network) with vertices (neural units) i, $i = 1, \ldots, N$ and edges (connections) defined by an adjacency matrix* $\mathbf{A}$, $A_{ij} \in \{0, 1\}$. *Let* $M \subseteq \Re^D$ *be a given manifold of a D-dimensional embedding space and* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of pointers* $\mathbf{w}_i \in M$ *each of which is attached to a vertex i of the graph G. The graph G with its vertices i assigned to the locations* $\mathbf{w}_i$ *on M forms a perfectly topology preserving map of M, iff the graph G is the induced Delaunay triangulation* $\mathcal{D}_S^{(M)}$ *of S.*

Proof. The proof is straightforward. If G is the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ of S, vertices i, j that are adjacent in G ($A_{ij} = 1$) are assigned to masked Voronoi polyhedra $V_i^{(M)}, V_j^{(M)}$ that are neighboring on M, and, vice versa, masked Voronoi polyhedra $V_i^{(M)}, V_j^{(M)}$ that are neighboring on M belong to vertices i, j that are adjacent in G. Then the mapping Φ_S from M to G as well as the inverse mapping Φ_S^{-1} from G to M are neighborhood preserving, and, hence, the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ of S forms a perfectly topology preserving map of M.

If the graph G forms a perfectly topology preserving map of M, then the mapping Φ_S from M to G as well as the inverse mapping Φ_S^{-1} from G to M are neighborhood preserving. This is valid if vertices i, j that are adjacent in G ($A_{ij} = 1$) are assigned to masked Voronoi polyhedra $V_i^{(M)}, V_j^{(M)}$ that are neighboring on M, and if masked Voronoi polyhedra $V_i^{(M)}, V_j^{(M)}$ that are neighboring on M belong to vertices i, j that are adjacent in G. According to the definition, G is the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ of S. ■

3.2. Competitive Hebbian Rule Forms Perfect Topology Preserving Maps

The competitive Hebb rule constructs the full Delaunay triangulation of a set of points $\mathbf{w}_1, \ldots, \mathbf{w}_N$ only if each Voronoi polyhedron of second-order V_{ij} is, at least partially, covered by the density distribution $P(\mathbf{v})$. If we define the manifold M as being the manifold of $\Re^D$ on which $P(\mathbf{v})$ is nonzero, two units i, j become connected iff $V_{ij} \cap M \neq \emptyset$. Hence, if the manifold M forms a submanifold that does not cover each Voronoi polyhedron of second order, the Delaunay triangulation will evolve only partly. If the distribution of the points $\mathbf{w}_i$ is dense on M in a sense we will define below, the part of the Delaunay triangulation that is formed by the competitive Hebb rule is the induced Delaunay triangulation and, hence, provides a perfectly topology preserving map of M.

DEFINITION 6. *Let* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of points* $\mathbf{w}_i$ *that are distributed over a given manifold* $M \subseteq \Re^D$. *The distribution of the points* $\mathbf{w}_i \in M$, $i = 1, \ldots, N$ *is dense on M, if for each* $\mathbf{v} \in M$ *the triangle* $\Delta(\mathbf{v}, \mathbf{w}_{i_0}, \mathbf{w}_{i_1})$ *formed by the point* $\mathbf{w}_{i_0}$ *that is closest to* $\mathbf{v}$, *the point* $\mathbf{w}_{i_1}$ *that is second closest to* $\mathbf{v}$, *and* $\mathbf{v}$ *itself lies completely on M, that is, if* $\Delta(\mathbf{v}, \mathbf{w}_{i_0}, \mathbf{w}_{i_1}) \subseteq M$ *is valid.*

A distribution of points $\mathbf{w}_i$ is dense on M according to the above definition, if the distribution is dense compared to the structural details of M. A dense distribution of the points $\mathbf{w}_i$ resolves the topological structure of M. If for each sample point $\mathbf{v} \in M$ there is a closest point $\mathbf{w}_{i_0}$ and a second closest point $\mathbf{w}_{i_1}$ such, that the triangle $\Delta(\mathbf{v}, \mathbf{w}_{i_0}, \mathbf{w}_{i_1})$ lies completely on M, the distribution of the $\mathbf{w}_i$'s is dense on M according to Definition 6. If the distribution is homogeneous, the distribution becomes dense simply by increasing the number N of points $\mathbf{w}_i$.

With Definition 6 we obtain the main theorem of this section:

THEOREM 3. *Let* $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ *be a set of points* $\mathbf{w}_i$ *that are distributed over a given manifold M. If the distribution of the points* $\mathbf{w}_i \in M$ *is dense on M, then the graph G that is formed by the competitive Hebb rule is the induced Delaunay triangulation* $\mathcal{D}_S^{(M)}$ *of S and, hence, forms a perfectly topology preserving map of M.*

Proof. Analog to Theorem 1 we prove the above theorem by showing that

$$V_i^{(M)} \cap V_j^{(M)} \neq \emptyset \leftrightarrow V_{ij}^{(M)} \neq \emptyset \qquad (16)$$

is valid, with $V_{ij}^{(M)} = V_{ij} \cap M$ as the masked Voronoi polyhedron of second order.

If $V_i^{(M)} \cap V_j^{(M)} \neq \emptyset$ is valid, there is a $\mathbf{v} \in M$ with $\mathbf{v} \in V_i$ and $\mathbf{v} \in V_j$. Then we obtain $\|\mathbf{v} - \mathbf{w}_i\| = \|\mathbf{v} - \mathbf{w}_j\| \leq \|\mathbf{v} - \mathbf{w}_k\|$ for all $\mathbf{w}_k \in S$, and, therefore, $\mathbf{v} \in V_{ij}^{(M)}$ is valid.

If $V_{ij}^{(M)} \neq \emptyset$ is valid, there is a $\mathbf{v}^* \in V_{ij}^{(M)}$ with $\Delta(\mathbf{v}^*, \mathbf{w}_i, \mathbf{w}_j) \subseteq M$, because the distribution of the pointers $\mathbf{w}_1, \ldots, \mathbf{w}_N$ is dense on M. For each $\mathbf{v} \in V_{ij}^{(M)}$ the points $\mathbf{w}_i$ and $\mathbf{w}_j$ are the two nearest neighbors. Without loss of generality we assume that for $\mathbf{v}^*$ the point $\mathbf{w}_i$ is the nearest neighbor. Because for each $\mathbf{u} \in \overline{\mathbf{v}^*\mathbf{w}_j}$ the point $\mathbf{w}_j$ is either the nearest or the second nearest neighbor of $\mathbf{u}$, and because for $\mathbf{u} = \mathbf{v}^*$ the point $\mathbf{w}_i$ is closest and for $\mathbf{u} = \mathbf{w}_j$ the point $\mathbf{w}_j$ is closest to $\mathbf{u}$, there is a $\mathbf{u}^* \in \overline{\mathbf{v}^*\mathbf{w}_j}$ for which $\|\mathbf{u}^* - \mathbf{w}_i\| = \|\mathbf{u}^* - \mathbf{w}_j\|$ is valid. Hence, we obtain $\mathbf{u}^* \in V_i$, $\mathbf{u}^* \in V_j$, and $\mathbf{u}^* \in \Delta(\mathbf{v}^*, \mathbf{w}_i, \mathbf{w}_j) \subseteq M$, which yields $M \cap (V_i \cap V_j) \neq \emptyset$ or, equivalently, $V_i^{(M)} \cap V_j^{(M)} \neq \emptyset$. ∎

With the theorem above we have shown that the competitive Hebb rule forms perfectly topology preserving maps, supposed the distribution of the points $\mathbf{w}_i$ is dense on the given manifold. In Figures 6 and 7 we show two simulation examples. In Figure 6 the manifold M consists of a three-dimensional (right parallelepiped), a two-dimensional (rectangle), and a one-dimensional (circle and connecting line) submanifold. Initially, the pointers $\mathbf{w}_i$ are distributed randomly within the space that embeds M, that is, $\mathbf{w}_i \in M$ is not valid necessarily. To obtain $\mathbf{w}_i \in M$ for each i, $i = 1, \ldots, N$ the pointers are distributed by employing the *neural gas* algorithm. The neural gas algorithm is an efficient, pattern driven vector quantization procedure and leads to a homogeneous distribution of the pointers $\mathbf{w}_i$ on M (Martinetz & Schulten, 1991; Martinetz, Berkovich, & Schulten, 1993).

Simultaneously to distributing the pointers the connections are formed. With each presentation of an input pattern $\mathbf{v} \in M$ a cycle of the competitive Hebb rule as well as of the neural gas algorithm is performed. Figure 6 shows the initial state, the network after 5000, 10,000, 15,000, 25,000, and at the final state after 40,000 adaptation steps. The pointers $\mathbf{w}_i$ change their locations slowly but permanently and, therefore, pointers that are neighboring at an early stage of the adaptation procedure might not be neighboring anymore at a more advanced stage. Those connections that were formed in the early stage and are not valid anymore in a later stage should die out. For that reason the competitive Hebb rule not only establishes but also refreshes connections. Connections that have not been refreshed for a while die out and are removed. The algorithm that results from combining the competitive Hebb rule with the neural gas algorithm is listed in Section 5.

At the end of the adaptation procedure, after the random presentation of 40,000 input patterns $\mathbf{v} \in M$, the connectivity structure corresponds to the induced Delaunay triangulation of the final pointer distribution and forms a perfectly topology preserving map that reflects the dimensionality and topological structure of the given manifold M.

In Figure 7 the manifold M is a torus. In this simulation example the pointers were distributed over M in a preprocessing stage, again by the neural gas algorithm. After having distributed the pointers, the connectivity structure is formed by the competitive Hebb rule. In contrast to the simulation example shown in Figure 6, the pointers stay fixed at their locations while the connections are formed. Therefore, each connection is definite and does not have to be refreshed. Depicted is the final result. Again, as in Figure 6, the connectivity structure forms a perfectly topology preserving map and reflects the dimensionality and topological structure of the manifold M.

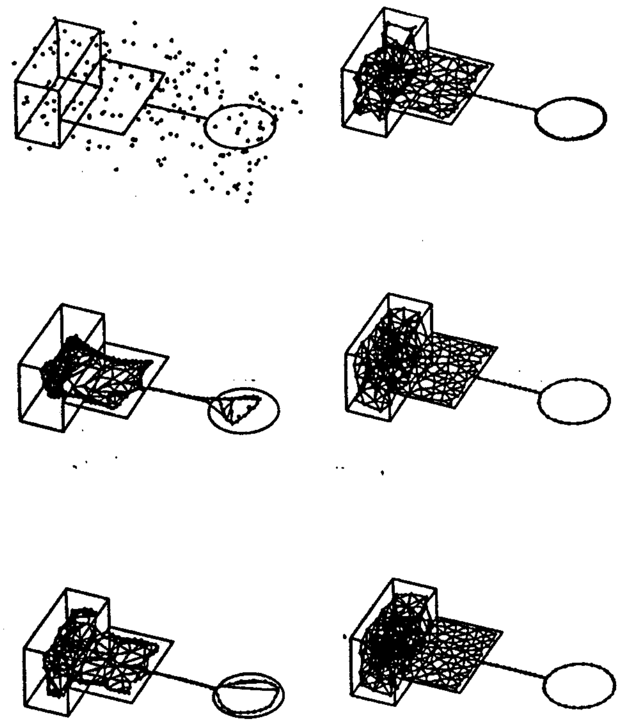


FIGURE 6. The competitive Hebb rule together with the neural gas algorithm forming a topology preserving map of a topologically heterogeneously structured manifold. The given manifold M consists of a three-dimensional (right parallelepiped), a two-dimensional (rectangle), and a one-dimensional (circle and connecting line) subset. The neural gas algorithm as an efficient input driven vector quantization procedure distributes the pointers $\mathbf{w}_i$ over the manifold M. With each presented pattern $\mathbf{v} \in M$ the competitive Hebb rule establishes or refreshes an edge of the induced Delaunay triangulation. Depicted are the initial state, the network after 5000, 10,000, 15,000, 25,000, and at the final state after 40,000 adaptation steps (from top left to bottom right). At the end of the adaptation procedure the network (graph) forms a perfectly topology preserving map that reflects the topological structure and the dimensionality of the manifold M.

4. COMPETITIVE HEBBIAN RULE FORMS PATH PRESERVING REPRESENTATIONS OF MANIFOLDS

If the distribution of the set S of points $\mathbf{w}_i$ is dense on the given manifold M, the competitive Hebb rule forms the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ and, hence, a perfectly topology preserving map of M, as stated in Theorem 3. If the distribution of the points $\mathbf{w}_i$ is dense on M, the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ has

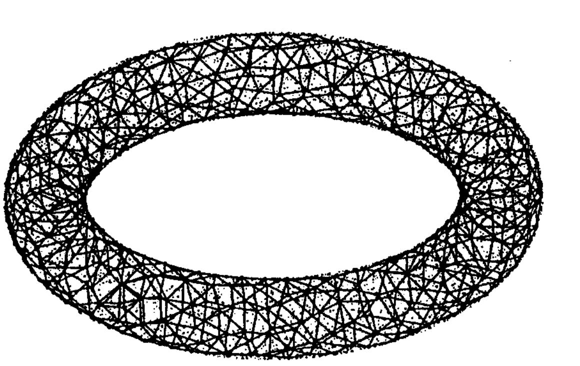


FIGURE 7. A topology preserving map of a torus, formed by the competitive Hebb rule. The pointers, the locations of which are marked by the large dots, were distributed over the given manifold M, that is, the torus, by the neural gas algorithm in a preprocessing stage. Then the pointers stay fixed and the edges are formed by the competitive Hebb rule. The small dots depict already presented elements of M. The few edges of the induced Delaunay triangulation that are still missing have small masked Voronoi polyhedra of second order and would emerge if further input patterns were presented.

the additional property that each edge $\overline{\mathbf{w}_i\mathbf{w}_j} \subseteq \Re^D$ of $\mathcal{D}_S^{(M)}$ lies completely on M. The union of all edges

$$\tilde{M}_S = \{\mathbf{v} \in \Re^D | \mathbf{v} \in \overline{\mathbf{w}_i\mathbf{w}_j} \text{ with } A_{ij} = 1\} \tag{17}$$

is a subset of M, that is, $\tilde{M}_S \subseteq M$ is valid. This property makes the competitive Hebb rule suited for forming path preserving representations of given manifolds M. With the following theorem we show that $\tilde{M}_S$ reflects the topology and forms a path preserving representation of M in the sense that two points $\mathbf{w}_i$, $\mathbf{w}_j$ are connected by a sequence of edges $\overline{\mathbf{w}_i\mathbf{w}_k}$, $\overline{\mathbf{w}_k\mathbf{w}_l}$, . . . , $\overline{\mathbf{w}_m, \mathbf{w}_j}$ of $\tilde{M}_S$, iff the points $\mathbf{w}_i$, $\mathbf{w}_j$ can be connected by a path on M.

THEOREM 4. *Let S be a set of points $\mathbf{w}_i$ that are distributed densely over a given manifold M, and let A_{ij} denote the elements of the adjacency matrix* A *of the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$. The union of all edges of the induced Delaunay triangulation is denoted by*

$$\tilde{M}_S = \{\mathbf{v} \in \Re^D | \mathbf{v} \in \overline{\mathbf{w}_i\mathbf{w}_j} \textit{ with } A_{ij} = 1\}. \tag{18}$$

$\tilde{M}_S$ forms a path preserving representation of M in the sense that two points $\mathbf{w}_i$, $\mathbf{w}_j$ of M are connected within $\tilde{M}_S$, iff they can be connected within M.

Proof. From $\tilde{M}_S \subseteq M$ it follows directly that two points $\mathbf{w}_i$, $\mathbf{w}_j$ that are connected within $\tilde{M}_S$ can also be connected within M.

The union of all Voronoi polyhedra V_i forms the embedding space $\Re^D$, that is, $\cup_{i=1}^N V_i = \Re^D$. Therefore, a path $P(\mathbf{w}_i, \mathbf{w}_j) \subseteq M$ that connects the points $\mathbf{w}_i$, $\mathbf{w}_j$ runs through a sequence $V_{i_0} = V_i, V_{i_1}, V_{i_2}, \ldots, V_{i_L} = V_j$ of $L + 1$ neighboring Voronoi polyhedra; that is, there is a sequence of Voronoi polyhedra $V_{i_0}, \ldots, V_{i_L}$, $L \geq 1$, with $V_{i_0} = V_i$, $V_{i_L} = V_j$, for which $V_{i_{l-1}} \cap V_{i_l} \cap P(\mathbf{w}_i, \mathbf{w}_j) \neq \emptyset$ for each $l = 1, \ldots, L$. From $P(\mathbf{w}_i, \mathbf{w}_j) \subseteq M$ and $V_{i_{l-1}} \cap V_{i_l} \cap P(\mathbf{w}_i, \mathbf{w}_j) \neq \emptyset$ for each $l = 1, \ldots, L$ follows $V_{i_{l-1}}^{(M)} \cap V_{i_l}^{(M)} \neq \emptyset$ for each $l = 1, \ldots, L$. Hence, there is a sequence of edges $\overline{\mathbf{w}_{i_0}\mathbf{w}_{i_1}}$, $\overline{\mathbf{w}_{i_1}\mathbf{w}_{i_2}}$, . . . , $\overline{\mathbf{w}_{i_{L-1}}\mathbf{w}_{i_L}}$ within M_S that connects $\mathbf{w}_i$ and $\mathbf{w}_j$. ■

Each path $P(\mathbf{w}_i, \mathbf{w}_j)$ on the manifold M between two points $\mathbf{w}_i$, $\mathbf{w}_j \in M$ can be mapped onto a best matching path $P'(\mathbf{w}_i, \mathbf{w}_j)$ within $\tilde{M}_S$, and, vice versa, each path $P(\mathbf{w}_i, \mathbf{w}_j)$ within $\tilde{M}_S$ is also a path between $\mathbf{w}_i$ and $\mathbf{w}_j$ on M. The path preserving representation $\tilde{M}_S$ of the manifold M not only yields a discretization of M, but also a discretization of the set of all paths on M. This allows the representation, classification, and planning of paths on M.

Figure 8 shows an example. In this computer simulation a "mouse" explores its environment that consists of the square area with a number of obstacles in it. While exploring the obstacle-free area by random walk, the "mouse" establishes a path preserving representation $\tilde{M}_S$ (which always is a perfectly topology preserving map) of its environment by employing the competitive Hebb rule. At each time step the "mouse's" current location $\mathbf{v}$, a two-dimensional vector, forms the input signal for the network. The pointers $\mathbf{w}_i$ attached to the units (vertices) of a network (graph) are distributed over the given manifold M, that is, the obstacle-free area (again by employing the neural gas algorithm). The procedure for distributing the pointers and simultaneously forming the connections is the same as the one employed in Figure 6 and is described in Section 5. The neural gas algorithm leads to a homogeneous distribution of the pointers $\mathbf{w}_i$ on M. Because the number of vertices is sufficient for obtaining a dense distribution, at the end of the exploration procedure the connectivity structure of the network (graph) corresponds to the induced Delaunay triangulation that defines both a perfectly topology preserving map and a path preserving representation $\tilde{M}_S$ of the obstacle-free area M. Firstly, each edge $\overline{\mathbf{w}_i\mathbf{w}_j}$ of the connectivity structure, that is, of the path preserving representation $\tilde{M}_S$, lies completely within M, that is, within the obstacle-free area. Secondly, any pair of points $\mathbf{w}_i$, $\mathbf{w}_j$ is connected within $\tilde{M}_S$ by a sequence of edges $\overline{\mathbf{w}_i\mathbf{w}_k}$, $\overline{\mathbf{w}_k\mathbf{w}_l}$, . . . , $\overline{\mathbf{w}_m, \mathbf{w}_j}$, iff the points $\mathbf{w}_i$, $\mathbf{w}_j$ can be connected by a path on M, that is, a path through the obstacle-free area. Thirdly, if a path between two points $\mathbf{w}_i$, $\mathbf{w}_j$ is short on $\tilde{M}_S$, it is also short on M. Because of these properties of the path preserving representation $\tilde{M}_S$, the network forms a representation or map of the obstacle-free area that can be employed by the "mouse" for planning short paths to target locations, as depicted in Figure 8.

5. TOPOLOGY REPRESENTING NETWORK (TRN)

This section presents the algorithm that has been employed in the computer simulations of Figures 6 and 8. This algorithm combines the neural gas algorithm (Martinetz & Schulten, 1991; Martinetz et al., 1993), for distributing the pointers $\mathbf{w}_i$, with the competitive

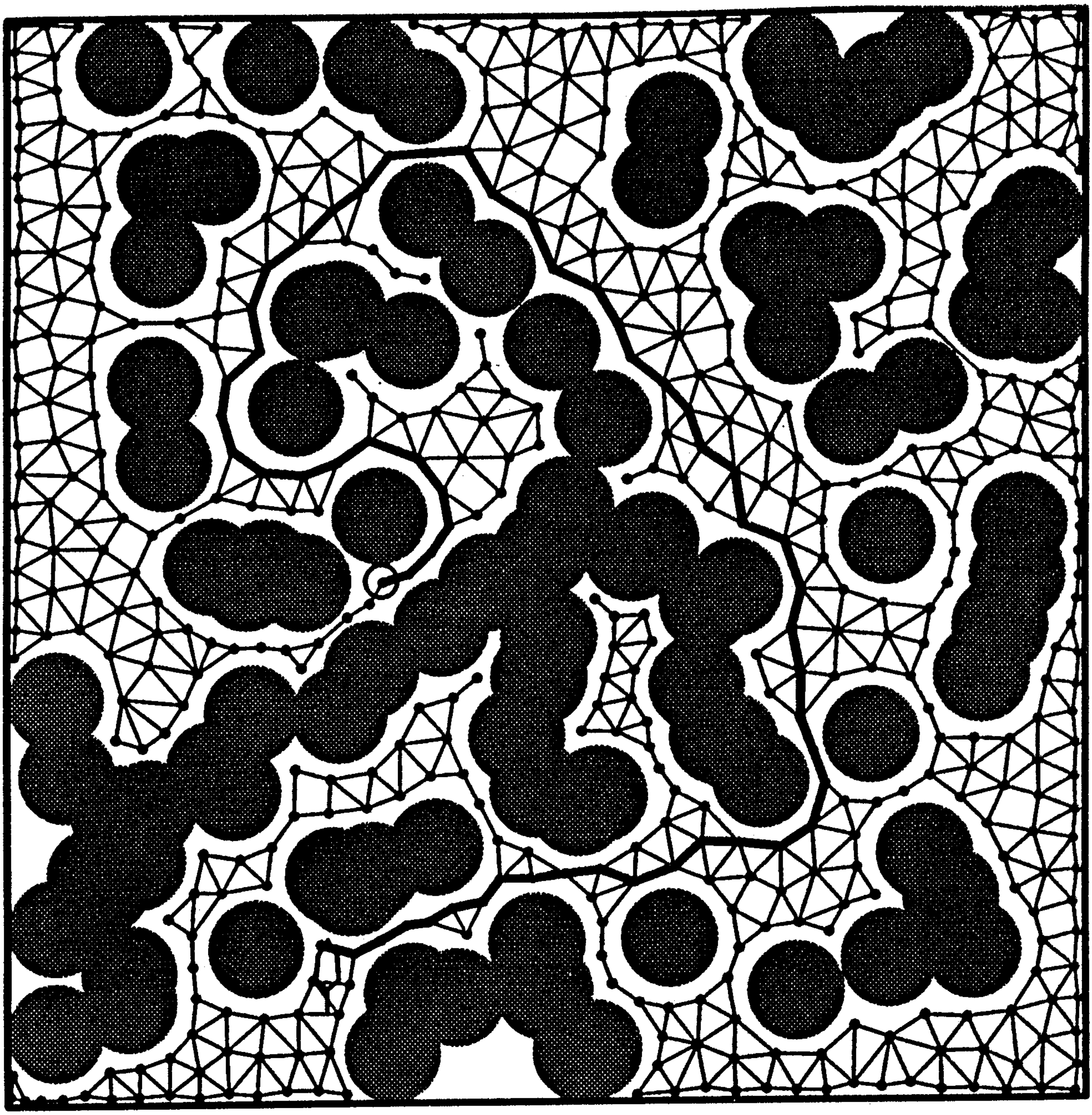


FIGURE 8. A "mouse" (lower left corner) and the discrete, path preserving representation of the environment it has formed by employing the competitive Hebb rule together with the neural gas algorithm. The dark areas are obstacles the "mouse" had to circumvent while exploring its environment by random walk. Successive positions of the "mouse" formed the input patterns for the network. The distribution of the pointer positions w_i, which are marked as dots, is dense. Hence, the connectivity structure that is formed by the competitive Hebb rule corresponds to the induced Delaunay triangulation and defines both a perfectly topology preserving map and a path preserving representation $\tilde{M}_s$ of the given manifold M, that is, the obstacle-free area.[4] The path preserving representation $\tilde{M}_s$ reflects the topology of the obstacle-free area and enables the "mouse" to plan short paths to target locations, for example, the location marked by the circle.

Hebb rule, for forming the connections. Simultaneously to distributing homogeneously the pointers w_i over the manifold M, the following procedure forms the induced Delaunay triangulation of the pointer distribution. Based on the results of the previous section we call this procedure together with the resulting network *topology representing network* or TRN.

Given is a manifold $M \subseteq \Re^D$ and a set of neural units i, $i = 1, \ldots, N$. To each unit i a pointer $w_i \in \Re^D$ is attached. The $N \times N$ connection strength matrix C, $C_{ij} \in \Re_0^+$, describes the connections between the units i, j. If $C_{ij} > 0$, there is a connection $i - j$ between unit i and unit j, and if $C_{ij} = 0$, there is no connection between i and j. Each connection $i - j$ has an age t_{ij}

[4] A few edges are still missing because the corresponding masked Voronoi polygons of second order have a small volume and, therefore, have not been hit yet by an input pattern. Notice that the "mouse" has a nonvanishing volume and, therefore, is not able to walk through very narrow openings.

that is the number of adaptation steps t the connection already exists without having been refreshed. If the age of a connection exceeds its lifetime T, the connection is removed. Connections have to die out because pointers that are neighboring at an early stage of the adaptation procedure might not be neighboring anymore at a more advanced stage. However, a connection does not die out if it is regularly refreshed by the competitive Hebb rule. This yields the following algorithm (Martinetz & Schulten, 1991):

(i) assign initial values to the pointers $w_i \in R^D$, $i = 1, \ldots, N$ and set all connection strengths C_{ij} to zero;

(ii) select an input pattern $v \in M$ with equal probability for each v;

(iii) for each unit i determine the number k_i of units j with

$$\|v - w_j\| < \|v - w_i\|$$

by, for example, determining the sequence $(i_0, i_1, \ldots, i_{N-1})$ with

$$\|v - w_{i_0}\| < \|v - w_{i_1}\| < \ldots < \|v - w_{i_{N-1}}\|;$$

(iv) perform an adaptation step of the pointers w_i according to the neural gas algorithm by setting

$$w_i^{new} = w_i^{old} + \varepsilon \cdot e^{-k_i/\lambda}(v - w_i^{old}), \quad i = 1, \ldots, N;$$

(v) if $C_{i_0 i_1} = 0$, set $C_{i_0 i_1} > 0$ and $t_{i_0 i_1} = 0$, that is, connect i_0 and i_1. If $C_{i_0 i_1} > 0$, set $t_{i_0 i_1} = 0$, that is, refresh connection $i_0 - i_1$;

(vi) increase the age of all connections of i_0 by setting $t_{i_0 j} = t_{i_0 j} + 1$ for all j with $C_{i_0 j} > 0$;

(vii) remove those connections of unit i_0 the age of which exceeds T by setting $C_{i_0 j} = 0$ for all j with $C_{i_0 j} > 0$ and $t_{i_0 j} > T$; continue with (ii).

In the simulations of Figures 6 and 8 the parameter λ, the step size ε, and the lifetime T were dependent on the number of adaptation steps t already performed. This time dependence had the same form for all three parameters and was $g(t) = g_i(g_f/g_i)^{t/t_{max}}$ [for example, $\lambda(t) = \lambda_i(\lambda_f/\lambda_i)^{t/t_{max}}$] with $\lambda_i = 0.2N$, $\lambda_f = 0.01$, $\varepsilon_i = 0.3$, $\varepsilon_f = 0.05$, $T_i = 0.1N$, $T_f = 2N$, and $t_{max} = 200N$. In the simulation of Figure 6 the number N of pointers w_i was 200, and in the simulation of Figure 8 the number N was 500.

6. SUMMARY AND CONCLUSION

We showed that formal neural units i form connectivity structures corresponding to Delaunay triangulations, if the Hebb rule together with competition among the connections is employed. Each neural unit i has to have a localized receptive field within the feature space M. By sequentially presenting patterns $v \in M$ and each time connecting those two units i, j that have the highest correlated output activity $y_i \cdot y_j$, the Delaunay triangulation $\mathcal{D}_S$ of the receptive field centers w_i evolves. Delaunay triangulations play an important role in a variety of information processing tasks. These tasks range from speech and image processing to robot control and the efficient storage and transfer of data. We demonstrated that the Delaunay triangulation with its significant role for information processing can be established by a self-organizing neural network: a set of formal neural units with lateral connections formed by an input driven, Hebbian learning rule.

In many information processing tasks, neighborhood and topological relations between patterns have to be exploited. Topology preserving maps provide pattern representations that are suitable for these tasks. Topology preserving maps can be found in various parts of the nervous system, and as artificial neural network models they have found widespread application (see, e.g., Kohonen, 1990). Existing models of topology preserving maps, however, are not able to provide perfect neighborhood and topology preservation if the topological structure of the pattern manifold is not known *a priori* or is intricate and not simply one-, two-, three-dimensional etc. In fact, the most common approaches choose *a priori* a graph that represents topological relationships, for example, a two-dimensional grid, and seek then the best match to the given pattern manifold. The approach presented in this paper rather determines such a graph, namely, an induced Delaunay triangulation, through matching to the given pattern manifold. This promises significant improvements in all applications where perfectly topology preserving pattern representations are essential but cannot be achieved with predetermined graphs.

In a first part of this paper we showed how the terms neighborhood preserving mapping and topology preserving map can be defined rigorously based on Voronoi polyhedra. For defining these two terms, we introduced masked Voronoi polyhedra and induced Delaunay triangulations. Both the masked Voronoi polyhedra and the induced Delaunay triangulation of a set of points depend on the shape of the given feature manifold M. We showed that the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$ as a particular subgraph of the full Delaunay triangulation $\mathcal{D}_S$ is a graph that forms a perfectly topology preserving map of the manifold M. In a second part of this paper we demonstrated through computer simulations and proved that the competitive Hebb rule forms induced Delaunay triangulations and, hence, yields perfectly topology preserving maps of feature manifolds. Necessary is a distribution of the receptive field centers w_i of the neural units i that is dense enough to resolve the shape of the manifold M. If the manifold $M \subseteq \Re^D$ fills the embedding space $\Re^D$ completely, then the competitive Hebb rule forms the full Delaunay triangulation $\mathcal{D}_S$ as a perfectly topology preserving map of M. If M is only a submanifold of $\Re^D$, then the competitive Hebb rule forms a subgraph of $\mathcal{D}_S$, that is, the induced Delaunay triangulation $\mathcal{D}_S^{(M)}$, as a perfectly topology preserving map of M.

In applications to control problems each unit i of the network can be mapped into a space of elements

a_i that specify control actions; for example, a_i is a vector of parameters that specifies an affine map to generate particular postures of a robot arm (Ritter & Schulten, 1986; Martinetz et al., 1990; Ritter et al., 1992; Walter & Schulten, 1992; Hesselroth et al., 1993). The graph G defined by the induced Delaunay triangulation forms a perfectly topology preserving map of the manifold M of input patterns **v** that designate control tasks, for example, which designate desired arm postures. G allows one to determine the units that are assigned to adjacent parts of the state space M. G allows one to identify *all* the units that have to adapt to similar control tasks, specified by elements a_i that are close to each other in a respective metric space. This identification enables the network to perform cooperative learning by adapting concertedly those units that are connected within the graph G. Cooperative learning leads to a significant increase of the learning rate and to a significant improvement of the robustness of the learning procedure for the elements a_i (Ritter et al., 1992; Walter & Schulten, 1992; Hesselroth et al., 1993). In some applications the cooperation between adjacent units is even necessary for convergence (Martinetz & Schulten, 1992). The induced Delaunay triangulation as a perfectly topology preserving map allows one to exploit optimally the advantages of cooperative learning and cooperative execution.

When the receptive fields are dense enough to resolve the shape of the given manifold M, the competitive Hebb rule forms a graph G that not only defines a perfectly topology preserving map, but in addition can be taken as a discrete, path preserving model or representation of M. The graph G corresponds to a particular induced Delaunay triangulation, each edge of which lies within the manifold. As a skeleton-like model of M, the graph G not only provides a discrete representation of all elements of M, but also a discrete representation of all paths on M. We presented a compact algorithm that combines the competitive Hebb rule with an input driven vector quantization procedure, that is, the neural gas algorithm (Martinetz & Schulten, 1991; Martinetz et al., 1993), for adaptively forming induced Delaunay triangulations G as perfectly topology preserving maps and discrete, path preserving representations of given manifolds M. Such a graph G together with the procedure that forms G we call *topology representing network* (TRN).

Applications in which models or representations of manifolds play an important role are in clustering, system analysis, and process control. In clustering, employing the TRN allows one to determine which parts of a given pattern manifold are separated and, hence, form different clusters. This is a novel approach to the problem of clustering. In system analysis, the TRN as a model of the submanifold that is formed by all possible states of a system yields information about the inner degrees of freedom of the system in different parts of the state space. In process control, employing the TRN for modeling the submanifold of all allowed states of a plant allows one to check whether a given state lies at the border or even outside the submanifold of all allowed states and, hence, is faulty or at least close to becoming critical (see also Kasslin, Kangas, & Simula, 1992).

Each path on the given manifold M is represented by the best matching path within the TRN, and each path through the TRN is a path on M. Furthermore, paths that are short within the TRN are also short on the manifold M. This allows the description, classification, and planning of paths. An application is in speech recognition (Brandt et al., 1991), as we mentioned already in Section 3. Other applications are in trajectory formation of robot arms or in process control, where plants have to be taken from one state to another by going through allowed regions of the state space.

We finally like to comment on a possible analogy between the TRN graph and the architecture of biological neural networks. The analogy identifies the vertices of the graph G with neurons of a layer M', the pattern manifold M with a layer of receptive neurons, the mapping from M onto the vertices of G with neural projections from M to M', the Voronoi polyhedra with receptive fields of the neurons of M', and, finally, the edges of the graph G with lateral connections within the layer M'. From this analogy one can derive the conclusion that lateral connections within a neural network can serve to represent the neighborhood and topological relations between features detected in a sensory layer or between tasks executed in a motor layer, etc., and allow the brain to process typical sequences of features or motions by following the paths that are established by these lateral connections.

REFERENCES

Bauer, H. U., & Pawelzik, K. R. (1992). Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4), 570–579.

Blasdel, G. G., & Salama, G. (1986). Voltage-sensitive dyes reveal a modular organization in monkey striate cortex. *Nature*, **321**, 579–585.

Bollobás, X. (1979). Graph theory, an introductory course. New York: Springer-Verlag.

Brandt, W. D., Behme, H., & Strube, H. W. (1991). Bildung von Merkmalen zur Spracherkennung mittels phonotopischer Karten. In Fortschritte zur Akustik—DAGA 91, Bad Honnef, DPG-GMBH, Germany, pp. 1057–1060.

Cooper, L. N. (1973). A possible organization of animal memory and learning. In B. Lundqvist & S. Lundqvist (Eds.), *Collective properties of physical systems* (24th Nobel Symposium) (pp. 252–264). New York: Academic Press.

Dalger, P., Noël, B., Martinetz, T., & Schulten, K. (1992). *Delaunay triangulation and Voronoi polyhedra of second order*. (Tech. Rep. TB 98-01). University of Illinois at Urbana–Champaign, Beckman Institute.

Delaunay, B. (1934). Sur la Spère Vide. *Bulletin of the Academy of Sciences USSR*, VII, 793–800.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numer. Mathematics*, **1**(5), 269–271.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Durbin, R., & Mitchison, G. (1990). A dimension reduction framework for understanding cortical maps. *Nature*, **343**, 644–647.

Fritzke, B. (1991). Let it grow—self-organizing feature maps with problem dependent cell structure. In T. Kohonen et al. (Eds.), *Artificial neural networks* (Vol. I, pp. 403–408). Amsterdam: North Holland.

George, J. A. (1971). *Computer implementation of the finite element method* (Tech. Rep. STAN-CS-71-208). Stanford University, Computer Science Department.

Gower, J. C., & Ross, G. J. S. (1969). Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, **18**(1), 54–64.

Gray, R. M. (1984). Vector quantization. *IEEE ASSP*, **1**(2), 4–29.

Grossberg, S. (1974). Classical and instrumental learning by nonlinear neural networks. In E. Rosen & F. Snell (Eds.), *Progress in Theoretical Biology* (Vol 3, pp 51–141). New York: Academic Press.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.

Hebb, D. (1949). *Organization of behavior*. New York: Wiley.

Hesselroth, T., Salkar, K., van der Smagt, P., & Schulten, K. (1993). Neural network control of a pneumatic robot arm. *IEEE Transactions on Systems, Man and Cybernetics* (in press).

Hubel, D. H., & Wiesel, T. N. (1974). Sequence regularity and geometry of orientation columns in the monkey striate cortex. *Journal of Comparative Neurology*, **158**, 267–294.

Kaas, J. H., Nelson, R. J., Sur, M., Lin, C. S., & Merzenich, M. M. (1979). Multiple representations of the body within the primary somatosensory cortex of primates. *Science*, **204**, 521–523.

Kangas, J. A., Kohonen, T. K., & Laaksonen, J. T. (1990). Variants of self-organizing maps. *IEEE Transactions on Neural Networks*, **1**(1), 93–99.

Kasslin, M., Kangas, J., & Simula, O. (1992). Process state monitoring using self-organizing maps. In I. Aleksander & J. Taylor (Eds.), *Artificial neural networks* 2 (Vol. 2, pp. 1531–1534). Amsterdam: North Holland.

Kelso, S. R., Ganong, A. H., & Brown, T. H. (1986). Hebbian synapses in hippocampus. *Proceedings of the National Academy of Sciences USA*, **83**, 5326–5330.

Knudsen, E. I., du Lac, S., & Esterly, S. D. (1987). Computational maps in the brain. *Annual Review of Neuroscience*, **10**, 41–65.

Knuth, D. E. (1973). *The art of computer programming. Volume III: Sorting and searching*. Reading, MA: Addison-Wesley.

Kohonen, T. (1982a). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59–69.

Kohonen, T. (1982b). Analysis of a simple self-organizing process. *Biological Cybernetics*, **44**, 135–140.

Kohonen, T., Mäkisara, K., & Saramäki, T. (1984). Phonotopic maps—insightful representation of phonological features for speech recognition (pp. 182–185). *Proceedings 7th International Conference on Pattern Recognition, Montreal.*

Kohonen, T., Oja, E., & Lehtio, P. (1985). In W. B. Levy, J. A. Anderson, & S. Lehmkuhle (Eds.), Synaptic modification, neuron selectivity, and nervous system organization (pp. 209–222). Hillsdale, NJ: Erlbaum.

Kohonen, T. (1989). *Self-organization and associative memory* (3rd ed.). Heidelberg: Springer Series in Information Sciences 8.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, **78**, 1464–1480.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7, 48–50.

Makhoul, J., Roucos, S., & Gish, H. (1985). Vector quantization in speech coding. *Proceedings of the IEEE*, **73**, 1551–1588.

von der Malsburg, C., Willshaw, D. J. (1977). How to label nerve cells so that they can interconnect in an ordered fashion. *Proceedings of the National Academy of Sciences USA*, 74, 5176–5178.

Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology*, **202**, 437–470.

Martinetz, T., Ritter, H., & Schulten, K. (1990). Three-dimensional neural net for learning visuomotor-coordination of a robot arm. *IEEE Transactions on Neural Networks*, **1**(1), 131–136.

Martinetz, T., & Schulten, K. (1991). "Neural gas" network learns topologies. In T. Kohonen et al. (Eds.), *Artificial neural networks* (Vol. I, pp. 397–402). Amsterdam: North Holland.

Martinetz, T., Berkovich, S., & Schulten, K. (1993). "Neural-gas" network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4), 558–569.

Martinetz, T., & Schulten, K. (1992). A neural network for robot control: Cooperation between neurons as a requirement for learning. *Computers & Electrical Engineering*, **19**(4), 315–332.

Nasrabadi, N. M., & King, R. A. (1988). Image coding using vector quantization: A review. *IEEE Transactions on Communications*, **36**(8), 957–971.

Nasrabadi, N. M., & Feng, Y. (1988). Vector quantization of images based upon the Kohonen self-organizing feature maps. In: *IEEE International Conference on Neural Networks* (pp. 1101–1108). San Diego.

Naylor, J., & Li, K. P. (1988). Analysis of a neural network algorithm for vector quantization of speech parameters. In *Proceedings of the First Annual INNS Meeting* (p. 310). New York: Pergamon Press.

Omohundro, S. M. (1990). *The Delaunay triangulation and function learning* (Tech. Rep. TR-90-001). Berkeley, CA: International Computer Science Institute.

Osteen, R. E., & Lin, P. P. (1974). Picture skeletons based on eccentricities of points of minimum spanning trees. *SIAM Journal of Computing*, **3**(1), 23–40.

Palm, G. (1982). *Neural assemblies: An alternative approach to artificial intelligence*. New York: Springer.

Preparata, F. P., & Shamos, M. I. (1988). *Computational geometry: An introduction*. New York: Springer-Verlag.

Prim, R. C. (1957). Shortest connecting networks and some generalizations. *Bell System Technical Journal*, **36**, 1389–1401.

Ritter, H., & Schulten, K. (1986). Topology conserving mappings for learning motor tasks. In J. S. Denker (Ed.), *Neural Networks for Computing*. AIP Conference Proceedings, 151:376–380, Snowbird, Utah.

Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural computation and self-organizing maps*. Reading, MA: Addison-Wesley.

Rosenkrantz, D. J., Stearns, R. E., & Lewis, P. M. (1974). Approximate algorithms for the traveling salesperson problem. *Fifteenth Annual IEEE Symposium on Switching and Automata Theory, 33–42.*

Shamos, M. I. (1978). Computational geometry. Unpublished doctoral thesis, Department of Computer Science, Yale University.

Strang, G., & Fix, G. (1973). *An analysis of the finite element method*. Englewood Cliffs, NJ: Prentice-Hall.

Suga, N., & O'Neill, W. E. (1979). Neural axis representing target range in the auditory cortex of the mustache bat. *Science*, **206**, 351–353.

Takeuchi, A., & Amari, S. (1979). Formation of topographic maps and columnar microstructures. *Biological Cybernetics*, **35**, 63–72.

Walter, J., & Schulten, K. (1993). Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks*, **4**(1), 86–95.

Willshaw, D. J., & von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London, Section B*, **194**, 431–445.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computing*, **C-20**(1), 68–86.