Tristrips on Hopfield Networks

Šíma, Jiří

2004

Institute of Computer Science
Academy of Sciences of the Czech Republic

Tristrips on Hopfield networks

Jiří Šíma

Technical report No. 908

June 2004

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 053 030, fax: +420 286 585 789,
e-mail:sima@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Tristrips on Hopfield networks

Jiří Šíma[1]

Technical report No. 908

June 2004

Abstract:

The important task of generating the minimum number of sequential triangle strips (tristrips) for a given triangulated surface model is motived by applications in computer graphics. This hard combinatorial optimization problem is reduced to the minimum energy problem in Hopfield nets by a linear-size construction. In particular, the classes of equivalent optimal stripifications are mapped one to one to the minimum energy states that are reached during any sequential computation by a Hopfield network starting at the zero initial state. Thus the underlying Hopfield network powered by simulated annealing (i.e. Boltzmann machine) which is implemented in a program HTGEN can be used for computing the semi-optimal stripifications. Practical experiments confirm that one can obtain much better results using HTGEN than by a leading stripification program FTSG although the running time of simulated annealing grows rapidly when the global optimum is being approached.

Keywords:
sequential triangle strip, combinatorial optimization, Hopfield network, minimum energy, simulated annealing

[1]This work was partially supported by project LN00A056 of The Ministry of Education of the Czech Republic. I thank Joseph S.B. Mitchell for sending me the FTSG program.

1 Sequential triangular strips

Piecewise-linear surfaces defined by sets of triangles (triangulation) are widely used representations for geometric models. Computing a succinct encoding of a triangulated surface model represents an important problem in graphics and visualization. Current 3D graphics rendering hardware often faces a memory bus bandwidth bottleneck in the processor-to-graphics pipeline. Apart from reducing the number of triangles that must be transmitted it is also important to encode the triangulated surface efficiently. A common encoding scheme is based on sequential triangle strips which avoid repeating the vertex coordinates of shared triangle edges. Triangle strips are supported by several graphics libraries (e.g. IGL, PHIGS, Inventor, OpenGL).

In particular, a *sequential triangle strip* (hereafter briefly *tristrip*) of length $m-2$ is an ordered sequence of $m \geq 3$ vertices $\sigma = (v_1, \ldots, v_m)$ which encodes the set of $n(\sigma) = m-2$ different triangles $T_\sigma = \{\{v_p, v_{p+1}, v_{p+2}\}\,;\, 1 \leq p \leq m-2\}$ so that their shared edges follow alternating left and right turns as indicated in Figure 1.1.a by a dashed line. Thus a triangulation consisting of a single tristrip with n triangles allows transmitting of only $n+2$ (rather than $3n$) vertices. In general, a triangulated surface model T with n triangles that is decomposed into k tristrips $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ requires only $n+2k$ vertices to be transmitted. A crucial problem is to decompose a triangulated surface model into the fewest tristrips. This stripification problem has recently been proved to be NP-complete in article [3] which also contains a more detailed related discussion complemented by references. In the present paper a new method of generating tristrips Σ for a given triangulated surface model T with n triangles is proposed which is based on a linear-time reduction to the minimum energy problem in Hopfield network $\mathcal{H}_T$ having $O(n)$ units and connections. This approach has been inspired by a more complicated and incomplete reduction (sequential cycles were not excluded) introduced in [8] which was supported only by experiments.

The paper is organized as follows. After a brief review of the basic definitions concerning Hopfield nets in Section 2, the main construction of Hopfield network $\mathcal{H}_T$ for a given triangulation T is described in Section 3. The correctness of this reduction is formally verified in Section 4 by proving a one-to-one correspondence between the classes of equivalent optimal stripifications of T and the minimum energy states reached by $\mathcal{H}_T$ during any sequential computation starting at the zero initial state (or $\mathcal{H}_T$ can be initialized arbitrarily if one asymmetric weight is introduced). This provides another NP-completeness proof for the minimum energy problem in Hopfield nets. In addition, $\mathcal{H}_T$ combined with simulated annealing (i.e. Boltzmann machine) has been implemented in a program HTGEN which is compared against a leading stripification program FTSG in Section 5. Practical experiments show that HTGEN can compute much better stripifications than FTSG although the running time of HTGEN grows rapidly when the global optimum is being approached.

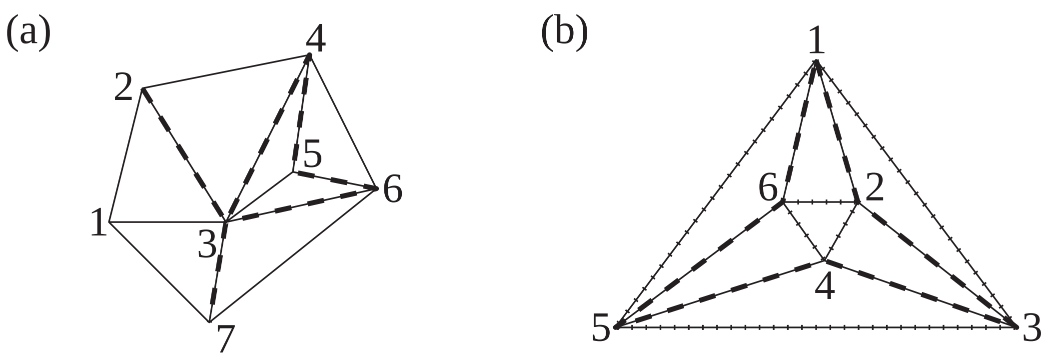

Figure 1.1: (a) Tristrip (1,2,3,4,5,6,3,7,1) (b) Sequential cycle (1,2,3,4,5,6,1,2)

2 The minimum energy problem

In his 1982 paper [5], John Hopfield introduced a very influential associative memory model which has since come to be widely known as the (symmetric) Hopfield network. The fundamental characteristic of this model is its well-constrained convergence behavior as compared to arbitrary asymmetric networks. Part of the appeal of Hopfield nets also stems from their connection to the much-studied Ising spin glass model in statistical physics [2], and their natural hardware implementations using electrical networks [6] or optical computers [4]. Hopfield networks have also been applied to the fast approximate solution of combinatorial optimization problems [7].

Formally, a Hopfield network is composed of s computational *units* or *neurons*, indexed as $1, \ldots, s$, that are connected into undirected graph or *architecture*, in which each connection between unit i and j is labeled with an integer symmetric *weight* $w(i,j) = w(i,j)$. The absence of a connection within the architecture indicates a zero weight between the respective neurons, and vice versa. For example, $w(j,j) = 0$ is assumed for $j = 1, \ldots, s$. The *sequential discrete* dynamics of such a network is here considered, in which the evolution of the network *state* $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_s^{(t)}) \in \{0,1\}^s$ is determined for discrete time instants $t = 0, 1, \ldots,$ as follows. The *initial state* $\mathbf{y}^{(0)}$ may be chosen arbitrarily, e.g. $\mathbf{y}^{(0)} = (0, \ldots, 0)$. At discrete time $t \geq 0$, the *excitation* of any neuron j is defined as $\xi_j^{(t)} = \sum_{i=1}^{s} w(i,j) y_i^{(t)} - h(j)$ including an integer *threshold* $h(j)$ local to unit j. At the next instant $t+1$, one (e.g. randomly) selected neuron j computes its new output $y_j^{(t+1)} = H(\xi_j^{(t)})$ by applying the Heaviside activation function H, that is, j is *active* when $H(\xi) = 1$ for $\xi \geq 0$ while j is *passive* when $H(\xi) = 0$ for $\xi < 0$. The remaining units do not change their states, i.e. $y_i^{(t+1)} = y_i^{(t)}$ for $i \neq j$. In this way the new network state $\mathbf{y}^{(t+1)}$ at time $t+1$ is determined.

In order to formally avoid long constant intermediate computations when only those units are updated that effectively do not change their outputs, a *macroscopic time* $\tau = 0, 1, 2, \ldots$ is introduced during which all the units in the network are updated. A computation of a Hopfield network *converges* or *reaches a stable state* $\mathbf{y}^{(\tau^*)}$ at macroscopic time $\tau^* \geq 0$ if $\mathbf{y}^{(\tau^*)} = \mathbf{y}^{(\tau^*+1)}$. The well-known fundamental property of a symmetric Hopfield network is that its dynamics is constrained by *energy* function

$$E(\mathbf{y}) = -\frac{1}{2} \sum_{j=1}^{s} \sum_{i=1}^{s} w(i,j) y_i y_j + \sum_{j=1}^{s} h(j) y_j \tag{2.1}$$

which is a bounded function defined on its state space whose value decreases along any nonconstant computation path (to be precise it is assumed here without loss of generality that $\xi_j^{(t)} \neq 0$). It follows from the existence of such a function that starting from any initial state the network converges towards some stable state corresponding to a local minimum of E [5]. Thus the cost function of a hard combinatorial optimization problem can be encoded into the energy function of a Hopfield network which is then minimized in the course of computation. Hence, the *minimum energy problem* of finding a network state with minimum energy is of special interest. Nevertheless, this problem is in general NP-complete [2] (see also [9] for related results).

A stochastic variant of Hopfield model called the *Boltzmann machine* [1] is also considered in which randomly selected unit j becomes active at time $t+1$, i.e. $y_j^{(t+1)} = 1$, with probability $P(\xi_j^{(t)})$ which is computed by applying the probabilistic activation function $P : \mathbf{R} \longrightarrow (0,1)$ defined as $P(\xi) = 1/(1 + e^{-2\xi/T^{(\tau)}})$ where $T^{(\tau)}$ is a so-called *temperature* at microscopic time $\tau > 0$. This parameter is controlled by *simulated annealing*, e.g.

$$T^{(\tau)} = \frac{T^{(0)}}{\log(1+\tau)} \tag{2.2}$$

for sufficiently high initial temperature $T^{(0)}$. The simulated annealing is a powerful heuristic method for avoiding the local minima in combinatorial optimization.

3 The reduction

For the purpose of reduction the following definition are introduced. Let T be a set of n triangles that represents a triangulated surface model homeomorphic to a sphere in which each edge is incident to at most two triangles. An edge is said to be *internal* if it is shared by exactly two triangles; otherwise it is a *boundary* edge. Denote by I and B the sets of internal and boundary edges, respectively, in triangulation T. Furthermore, a *sequential cycle* is a "cycled tristrip", that is, an ordered sequence of vertices $C = (v_1, \dots, v_m)$ where $m \geq 4$ is even, which encodes the set of $m-2$ different triangles $T_C = \{\{v_p, v_{p+1}, v_{p+2}\}\,;\, 1 \leq p \leq m-2\}$ so that $v_1 = v_{m-1}$ and $v_2 = v_m$. Also denote by I_C and B_C the sets of internal and boundary edges of sequential cycle C, respectively, that is $I_C = \{\{v_p, v_{p+1}\}\,;\, 1 \leq p \leq m-2\}$ and $B_C = \{\{v_p, v_{p+2}\}\,;\, 1 \leq p \leq m-2\}$. An example of the sequential cycle is depicted in Figure 1.1.b where its internal and boundary edges are indicated by dashed and dotted lines, respectively. In addition, let $\mathcal{C}$ be the set of all sequential cycles in T.

For each sequential cycle $C \in \mathcal{C}$ one unique *representative* internal edge $e_C \in I_C$ can be chosen as follows. Start with any cycle $C \in \mathcal{C}$ and choose any edge from I_C to be its representative edge e_C. Observe that for a fixed orientation of triangulated surface any internal edge follows either left or right turn corresponding to at most two sequential cycles. Thus denote by C' the sequential cycle having no representative edge so far which shares its internal edge $e_C \in I_C \cap I_{C'}$ with C if such C' exists; otherwise let C' be any sequential cycle with no representative internal edge or stop if all the sequential cycles do have their representative edges. Further choose any edge from $I_{C'} \setminus \{e_C\}$ to be the representative edge $e_{C'}$ of C' and repeat the previous step with C replaced by C'. Clearly, each edge represents at most one cycle because set $I_{C'} \setminus \{e_C\} \neq \emptyset$ always contains only edges that do not represent any cycle so far. If it were not the case then another sequential cycle C'' different from C would obtain its representative edge $e_{C''}$ from $I_{C'} \cap I_{C''}$ and hence a representative edge would already be assigned to C' (immediately after $e_{C''}$ was assigned to C'') before C is considered.

Hopfield network $\mathcal{H}_T$ corresponding to triangulation T will now be constructed. With each internal edge $e = \{v_1, v_2\} \in I$ two neurons ℓ_e and r_e are associated whose states either $y_{\ell_e} = 1$ or $y_{r_e} = 1$ indicate that e follows the left or right turn, respectively, along a tristrip according to the chosen orientation of triangulated surface. Let $L_e = \{e, e_1, e_2, e_3, e_4\}$ with $e_1 = \{v_1, v_3\}$, $e_2 = \{v_2, v_3\}$, $e_3 = \{v_2, v_4\}$, and $e_4 = \{v_1, v_4\}$ be the set of edges of the two triangles $\{v_1, v_2, v_3\}$, $\{v_1, v_2, v_4\}$ that share edge e. Denote by $J_e = \{\ell_f, r_f\,;\, f \in L_e \cap I\}$ the set of neurons local to e that are associated with the internal edges from L_e. Unit ℓ_e is connected with all neurons from J_e via negative weights except for units r_{e_2} (if $e_2 \in I$), ℓ_e, and r_{e_4} (if $e_4 \in I$) whose states may encode a tristrip that traverses edge e by the left turn. Such a situation (for $L_e \subseteq I$) is depicted in Figure 3.1.a where the edges shared by triangles within the tristrip together with associated active neurons r_{e_2}, ℓ_e, r_{e_4} are marked. Similarly, unit r_e is connected with neurons from J_e except for units ℓ_{e_1} (if $e_1 \in I$), r_e, and ℓ_{e_3} (if $e_3 \in I$) corresponding to the right turn. Thus define weights

$$\begin{aligned} w(i, \ell_e) &= -7 \quad \text{for } i \in J_{\ell_e} = J_e \setminus \{r_{e_2}, \ell_e, r_{e_4}\} \\ w(i, r_e) &= -7 \quad \text{for } i \in J_{r_e} = J_e \setminus \{\ell_{e_1}, r_e, \ell_{e_3}\} \end{aligned} \tag{3.1}$$

for each internal edge $e \in I$. Hence, the states of Hopfield network $\mathcal{H}_T$ with the negative symmetric weights which enforce locally the alternation of left and right turns encode tristrips. Furthermore, for each representative edge e_C ($C \in \mathcal{C}$) define $j_C = \ell_{e_C}$ if e_C follows the left turn along sequential cycle C or $j_C = r_{e_C}$ if e_C follows the right turn along C. Let $J = \{j_C\,;\, C \in \mathcal{C}\}$ be the set containing all such neurons whereas $J' = \{\ell_e, r_e \notin J\,;\, e \in I\}$ denotes its complement. The thresholds of neurons associated with internal edges are defined:

$$h(j) = \begin{cases} -5 + 2b_{e(j)} & \text{for } j \in J' \\ 1 + 2b_{e(j)} & \text{for } j \in J\,, \end{cases} \tag{3.2}$$

where $e(j) = e$ for $j \in \{\ell_e, r_e\}$ and $b_e = |\{C \in \mathcal{C}\,;\, e \in B'_C\}| \leq 2$ for $B'_C = B_C \setminus L_{e_C}$.

Nevertheless, Hopfield network $\mathcal{H}_T$ must also avoid the states encoding cycled strips of triangles around sequential cycles [3]. As follows from the analysis below such infeasible states would have less energy (2.1) than those encoding the optimal stripifications. For this purpose, two auxiliary neurons d_C, a_C are introduced for each sequential cycle $C \in \mathcal{C}$. Unit d_C computes the disjunction of outputs

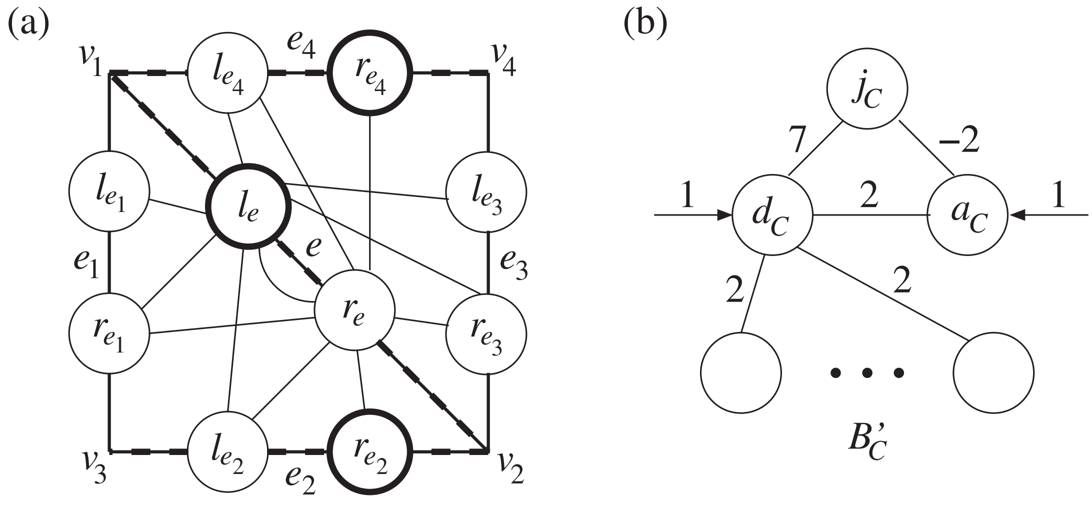

Figure 3.1: The construction of Hopfield network $\mathcal{H}_T$

from all neurons i associated with boundary edges $e(i) \in B'_C$ of C which, being active, enables the activation of unit j_C associated with representative edge e_C. Hence, any tristrip may pass through edge e_C along the direction of C only if a boundary edge of C is is a part of another tristrip crossing the sequential cycle C. This ensures that the states of Hopfield network $\mathcal{H}_T$ do not encode sequential cycles. In addition, unit a_C balances the contribution of d_C to the energy when j_C is passive. As depicted in Figure 3.1.b this is implemented by thresholds and symmetric weights:

$$\begin{aligned} h(d_C) &= h(a_C) = 1 && (3.3)\\ w(i, d_C) &= w(d_C, i) = 2 \quad \text{for } e(i) \in B'_C && (3.4)\\ w(d_C, j_C) &= w(j_C, d_C) = 7 && (3.5)\\ w(d_C, a_C) &= w(a_C, d_C) = 2\,, \quad w(j_C, a_C) = w(a_C, j_C) = -2 && (3.6) \end{aligned}$$

for each sequential cycle $C \in \mathcal{C}$. This completes the construction of Hopfield network $\mathcal{H}_T$.

Moreover, observe that the number of units $s = 2|I| + 2|\mathcal{C}|$ in $\mathcal{H}_T$ is linear in terms of triangulation size $n = |T|$ because the number of sequential cycles $|\mathcal{C}|$ can be upper bounded by $2|I| = O(n)$ since each internal edge can belong to at most two cycles. Similarly, the number of connections in $\mathcal{H}_T$ can be upper bounded by $7 \cdot 2|I| + 2 \cdot 2|I| + 3|\mathcal{C}| = O(n)$ according to (3.1) and (3.4)–(3.6) since again each internal edge may appear in B_C for at most two $C \in \mathcal{C}$. It can be checked that the reduction can also be done within linear time $O(n)$.

4 The correctness

The correctness of the reduction introduced in Section 3 will be verified by proving Theorem 1 below. Let $\mathcal{S}_T$ be the set of optimal stripifications with the minimum number of tristrips for T. Define $\Sigma \in \mathcal{S}_T$ is *equivalent* with $\Sigma' \in \mathcal{S}_T$ if their corresponding tristrips encode the same sets of triangles, i.e. $\Sigma \sim \Sigma'$ iff $\{T_\sigma\,;\, \sigma \in \Sigma\} = \{T_{\sigma'}\,;\, \sigma' \in \Sigma'\}$. For example, two equivalent optimal stripifications may differ in a tristrip σ encoding triangles $T_\sigma = T_C$ of sequential cycle C which is split at two different positions. Moreover, let $[\Sigma]_\sim = \{\Sigma' \in \mathcal{S}_T\,;\, \Sigma' \sim \Sigma\}$ be the class of optimal stripifications equivalent with $\Sigma \in \mathcal{S}_T$ and denote by $\mathcal{S}_T/_\sim = \{[\Sigma]_\sim\,;\, \Sigma \in \mathcal{S}_T\}$ the partition of $\mathcal{S}_T$ into equivalence classes.

Theorem 1 *Let $\mathcal{H}_T$ be a Hopfield network corresponding to triangulation T with n triangles and denote by $Y^* \subseteq \{0,1\}^s$ the set of stable states that can be reached during any sequential computation by $\mathcal{H}_T$ starting at the zero initial state. Then each state $\mathbf{y} \in Y^*$ encodes a correct stripification $\Sigma_\mathbf{y}$ of T into k tristrips and has energy*

$$E(\mathbf{y}) = 5(k - n)\,. \tag{4.1}$$

In addition, there is a one-to-one correspondence between the classes of equivalent optimal stripifications $[\Sigma]_\sim \in \mathcal{S}_T/_\sim$ *having the minimum number of tristrips for* T *and the states in* Y^* *with minimum energy* $\min_{\mathbf{y}\in Y^*} E(\mathbf{y})$.

Proof: Stripification $\Sigma_\mathbf{y}$ is decoded from $\mathbf{y} \in Y^*$ as follows. Denote by $I_0 = \{e \in I\,;\, y_{\ell_e} = y_{r_e} = 0\}$ the set of internal edges $e \in I$ whose associated neurons ℓ_e, r_e are passive and let $I_1 = I \setminus I_0$ be its complement. Set $\Sigma_\mathbf{y}$ contains each ordered sequence $\sigma = (v_1, \ldots, v_m)$ of $m \geq 3$ vertices that encodes $n(\sigma) = m-2$ different triangles $\{v_p, v_{p+1}, v_{p+2}\} \in T$ for $1 \leq p \leq m-2$ such that their edges $e_0 = \{v_1, v_3\}$, $e_m = \{v_{m-2}, v_m\}$, and $e_p = \{v_p, v_{p+1}\}$ for $1 \leq p \leq m-1$ satisfy $e_0, e_1, e_{m-1}, e_m \in I_0 \cup B$ and $e_2, \ldots, e_{m-2} \in I_1$. Notice that $\sigma \in \Sigma_\mathbf{y}$ with $n(\sigma) = 1$ encodes a single triangle with all its edges in $I_0 \cup B$. It will be proved that $\Sigma_\mathbf{y}$ is a correct stripification of T.

It will first be observed that every neuron $j \in J \cup J'$ associated with an internal edge is passive if there is an active unit $i \in J_j$. This ensures that each $\sigma \in \Sigma_\mathbf{y}$ encodes a set of different triangles T_σ whose shared edges follow alternating left and right turns. It also follows that sets T_σ are pairwise disjoint for $\sigma \in \Sigma_\mathbf{y}$. In particular, for each unit $j \in J \cup J'$ the number of positive weights (3.4) contributing to its excitation ξ_j is at most $b_{e(j)} \leq 2$ which are subtracted within threshold $h(j)$ according to (3.2). Hence, if all units $i \in J_j$ are passive then $\xi_j \leq 5$ for $j \in J'$, and $\xi_j \leq 6$ for $j \in J$ which may include weight (3.5). Thus, an active unit $i \in J_j$ contributing to ξ_j via negative weight (3.1) makes unit j passive due to $\mathbf{y}$ is a stable state. Further, it must also be checked that stripification $\Sigma_\mathbf{y}$ covers all triangles in T, that is, $\bigcup_{\sigma\in\Sigma_\mathbf{y}} T_\sigma = T$. According to the definition of $\Sigma_\mathbf{y}$ it suffices to prove that there is no sequential cycle $C = (v_1, \ldots, v_m)$ such that $e_p = \{v_p, v_{p+1}\} \in I_1$ for all $p = 1, \ldots, m-2$. On the contrary suppose that such C exists which implies $(B_C \cap I) \subseteq I_0$. It follows that unit $j_C \in J$ associated with $e_C = e_q$ for some $1 \leq q \leq m-2$ could not be activated during sequential computation of $\mathcal{H}_T$ starting at the zero state, that is, $y_{j_C}^{(t)} = 0$ for $t \geq 0$ since its positive threshold $h(j_C)$ defined in (3.2) can be reached only by weight (3.5) from d_C. However, d_C computes the disjunction of outputs from neurons i for $e(i) \in B'_C \subseteq I_0$ according to (3.3) and (3.6) which are passive in the course of computation. Hence, $y_{d_C}^{(t)} = 0$ for $t \geq 0$ making also unit a_C passive. Thus $e_q \in I_0$ which is a contradiction. This completes the argument for $\Sigma_\mathbf{y}$ to be a correct stripification of T.

Furthermore, assume that $\Sigma_\mathbf{y}$ contains k tristrips. ¿From the definition of $\Sigma_\mathbf{y}$ each tristrip $\sigma \in \Sigma_\mathbf{y}$ is encoded using $n(\sigma) - 1$ edges from I_1. Hence,

$$|I_1| = \sum_{\sigma\in\Sigma_\mathbf{y}} (n(\sigma) - 1) = n - k \tag{4.2}$$

which equals the number of active units in $J' \cup J$. It will be shown that each active neuron $j \in J' \cup J$ is accompanied with a contribution of -5 to energy (2.1) which gives (4.1) according to (4.2). Assume that neuron $j \in J' \cup J$ is active which implies $y_i = 0$ for all units $i \in J_j$. Moreover, neuron j is connected to $b_{e(j)}$ units d_C for $e(j) \in B'_C$ which are active since the underlying disjunctions include active j. Consider first the case of active neuron $j \in J'$ which produces the following contribution to the energy:

$$-\frac{1}{2}b_{e(j)}w(d_C, j) - \frac{1}{2}b_{e(j)}w(j, d_C) + h(j) = -b_{e(j)}w(d_C, j) + h(j) = -5 \tag{4.3}$$

according to (2.1), (3.2), and (3.4). Similarly, active neuron $j_C \in J$ assumes active unit d_C and makes a_C passive due to (3.3) and (3.6), which contributes to the energy:

$$-b_{e(j_C)}w(j_C, d_C) - w(d_C, j_C) + h(j_C) + h(d_C) = -5\,. \tag{4.4}$$

On the other hand, unit a_C balances the contribution of active neuron d_C to the energy when j_C is passive, that is, $-w(a_C, d_C) + h(d_C) + h(a_C) = 0$ according to (3.3) and (3.6).

Finally, optimal stripification $\Sigma \in \mathcal{S}_T$ is encoded by state $\mathbf{y}$ of $\mathcal{H}_T$ so that $\Sigma \in [\Sigma_\mathbf{y}]_\sim$. Equivalent stripification $\Sigma' \sim \Sigma$ is used to determine state $\mathbf{y}$ such that $\Sigma_\mathbf{y} = \Sigma'$. For each tristrip $\sigma \in \Sigma$ that encodes triangles $T_\sigma = T_C$ of some $C \in \mathcal{C}$, define a corresponding tristrip $\sigma' = (v_1, \ldots, v_m) \in \Sigma'$ having $T_{\sigma'} = T_\sigma$ so that σ' starts with representative edge $e_C = \{v_1, v_2\}$. Then neuron ℓ_e or r_e from

$J' \cup J$ is active iff there is tristrip $\sigma = (v_1, \ldots, v_m) \in \Sigma'$ such that its edge $e = \{v_p, v_{p+1}\}$ for some $2 \le p \le m-2$ follows the left or right turn, respectively. In addition, unit d_C for $C \in \mathcal{C}$ is active iff there is active neuron $i \in J' \cup J$ for $e(i) \in B'_C$ while unit a_C is active iff d_C is active and j_C is passive. It follows that $\mathbf{y}$ is a stable state of $\mathcal{H}_T$. It must still be proved that $\mathbf{y}$ can be reached during sequential computation by $\mathcal{H}_T$ starting at the zero initial state, that is $\mathbf{y} \in Y^*$.

Define a directed graph $\mathcal{G} = (\mathcal{C}, \mathcal{A})$ whose vertices are sequential cycles $C \in \mathcal{C}$ and $(C_1, C_2) \in \mathcal{A}$ is an edge of $\mathcal{G}$ iff $e_{C_1} \in B'_{C_2}$. Let $\mathcal{C}' \subseteq \{C \in \mathcal{C}\,;\, y_{j_C} = 1\}$ be a subset of all the vertices $C \in \mathcal{C}$ with $y_{j_C} = 1$ that create directed cycles in $\mathcal{G}$. On the contrary suppose that units i are passive for all $e(i) \in \bigcup_{C \in \mathcal{C}'} B'_C \setminus E_{\mathcal{C}'}$ where $E_{\mathcal{C}'} = \{e_C\,;\, C \in \mathcal{C}'\}$. Observe that for each $C \in \mathcal{C}'$ units i for $e(i) \in B_C \cap L_{e_C}$ are also passive due to active j_C. Thus it seems that such a stable state could not be reached during any sequential computation by $\mathcal{H}_T$ starting at the zero initial state since neurons j_C, $C \in \mathcal{C}'$, can be activated only by units d_C, $C \in \mathcal{C}'$, whose activation depends only on active j_C, $C \in \mathcal{C}'$, in this case. Since $\Sigma_{\mathbf{y}}$ is the optimal stripification, the underlying tristrips follow internal edges of sequential cycles $C \in \mathcal{C}'$ as much as possible being interrupted only by edges from $\bigcup_{C \in \mathcal{C}'} B_C \setminus E_{\mathcal{C}'}$. In addition, any tristrip $\sigma \in \Sigma_{\mathbf{y}}$ crossing some sequential cycle $C_1 \in \mathcal{C}'$, that is $\emptyset \ne T_\sigma \cap T_{C_1} \ne T_{C_1}$, has one its end within this cycle C_1 because σ may enter C_1 only through its boundary edge $e_{C_2} \in B_{C_1}$ (i.e. $y_{j_{C_2}} = 1$) which is the only representative edge of sequential cycle $C_2 \in \mathcal{C}'$ that σ follows. It will be proved below that there exists sequential cycle $C \in \mathcal{C}'$ containing two tristrips $\sigma_1, \sigma_2 \in \Sigma_{\mathbf{y}}$, that is $T_{\sigma_1} \subseteq T_C$ and $T_{\sigma_2} \subseteq T_C$. Hence, stripification $\Sigma'_{\mathbf{y}}$ with fewer tristrips can be constructed from $\Sigma_{\mathbf{y}}$ by introducing only one tristrip $\sigma^* \in \Sigma'_{\mathbf{y}}$ such that $T_{\sigma^*} = T_C$ (e.g. $y_{j_C} = 0$) instead of the two tristrips $\sigma_1, \sigma_2 \in \Sigma_{\mathbf{y}}$ while any tristrip $\sigma \in \Sigma_{\mathbf{y}}$ crossing and thus ending within sequential cycle C is shortened to $\sigma' \in \Sigma'_{\mathbf{y}}$ so that $T_{\sigma'} \cap T_C = \emptyset$ which does not increase the number tristrips. This will be a contradiction with the assumption that $\Sigma_{\mathbf{y}}$ is the optimal stripification, and hence $\mathbf{y} \in Y^*$.

In order to prove that sequential cycle $C \in \mathcal{C}'$ containing two tristrips exists consider first an intersection of two cycles $C_1, C_2 \in \mathcal{C}'$ that does not contain their representative edges, that is $e_{C_1} \notin L_e$ and $e_{C_2} \notin L_e$ for some edge $e \in I_{C_1} \cap I_{C_2}$. Adopt the notation introduced in Figure 3.1.a so that $e_1, e, e_3 \in I_{C_1}$ and $e_2, e, e_4 \in I_{C_2}$. Let triangle $\{v_1, v_2, v_3\} \in T_\sigma$ be encoded by tristrip $\sigma \in \Sigma_{\mathbf{y}}$. If either unit ℓ_e or neuron r_e is active then σ encodes exactly two triangles $T_\sigma = \{\{v_1, v_2, v_3\}, \{v_1, v_2, v_4\}\} \subseteq T_{C_1} \cap T_{C_2}$ sharing edge $e = \{v_1, v_2\}$ since r_{e_2}, r_{e_4} and ℓ_{e_1}, ℓ_{e_3} are passive neurons associated with boundary edges $e_2, e_4 \in B_{C_1} \setminus \{e_{C_2}\}$ and $e_1, e_3 \in B_{C_2} \setminus \{e_{C_1}\}$, respectively. On the other hand, if both units ℓ_e, r_e are passive then at most one neuron, either r_{e_1} or ℓ_{e_2}, may be active provided that $e_1, e_2 \in I_{C_3}$ for some sequential cycle $C_3 \in \mathcal{C}'$ with representative edge $e_{C_3} = e_1$ or $e_{C_3} = e_2$, respectively. In this case, σ may encode one more triangle from either T_{C_1} or T_{C_2} sharing edge e_1 or e_2, which cannot be further extended since σ following C_3 crosses C_2 or C_1, respectively. Thus conclude that the intersection of $C_1, C_2 \in \mathcal{C}'$ produces one short tristrip $\sigma \in \Sigma_{\mathbf{y}}$ of length $n(\sigma) \le 2$ encoding one or two triangles, and $T_\sigma \subseteq T_{C_1}$ or $T_\sigma \subseteq T_{C_2}$.

Furthermore, consider the subset of sequential cycles $\mathcal{C}'' \subseteq \mathcal{C}'$ whose boundary edges form the exterior boundary of set of triangles $\bigcup_{C \in \mathcal{C}'} T_C$. The triangles whose edges form a connected part of this exterior boundary belonging to one sequential cycle $C_1 \in \mathcal{C}''$ are encoded by single tristrip $\sigma_1 \in \Sigma_{\mathbf{y}}$ such that $T_{\sigma_1} \subseteq T_{C_1}$ because $\Sigma_{\mathbf{y}}$ is the optimal stripification and there is no sequential cycle in $\mathcal{C}'$ crossing $C_1 \in \mathcal{C}''$. Moreover, there is a so-called *boundary* intersection of two sequential cycles $C_1, C_2 \in \mathcal{C}''$ when the exterior boundary passes from the part formed by C_1 to that formed by C_2. It follows that there are at least $|\mathcal{C}''|$ such intersections. Assume first that there is a boundary intersection of $C_1, C_2 \in \mathcal{C}''$ that does not contain e_{C_1}, e_{C_2}. Hence this intersection produces one short tristrip $\sigma \in \Sigma_{\mathbf{y}}$ of length $n(\sigma) \le 2$ such that e.g. $T_\sigma \subseteq T_{C_1}$, which together with $\sigma_1 \in \Sigma_{\mathbf{y}}$ satisfying also $T_{\sigma_1} \subseteq T_{C_1}$ gives $C = C_1$ containing two tristrips. In the opposite case, each boundary intersection of $C_1, C_2 \in \mathcal{C}''$ must contain just one representative edge e_{C_1} or e_{C_2} because there are only $|\mathcal{C}''|$ edges representing the sequential cycles in $\mathcal{C}''$. Clearly, the other non-boundary intersection of $C_1, C_2 \in \mathcal{C}''$ exists which cannot contain e_{C_1}, e_{C_2}, and hence $C \in \mathcal{C}'$ containing two tristrips can be found again.

Obviously, the class of equivalent optimal stripifications $[\Sigma_{\mathbf{y}}]_\sim$ with the minimum number of tristrips corresponds uniquely to the state $\mathbf{y} \in Y^*$ having minimum energy $\min_{\mathbf{y} \in Y^*} E(\mathbf{y})$ according to (4.1). □

Note that the reduction in Theorem 1 together with the fact that the optimal stripification problem is NP-complete [3] provides another NP-completeness proof for the minimum energy problem in

Hopfield networks (cf. [2, 9]). In addition, the restriction to the zero initial network states in Theorem 1 can be inconvenient e.g. for stochastic computation. Without this constraint, however, $\mathcal{H}_T$ may reach infeasible states. In particular, initially active unit j_C can activate d_C in spite of $y_i = 0$ for all $e(i) \in B'_C$, which admits sequential cycle C. Nevertheless, this can be secured by introducing asymmetric weight $w(d_C, j_C) = 7$ whereas $w(j_C, d_C) = 0$ (cf. (3.5)). This revision which is used for experiments in Section 5 does not break the convergence of $\mathcal{H}_T$ to states $\mathbf{y} \in Y^*$.

5 Experiments

A C++ program HTGEN has been created to automate the reduction from Theorem 1 including the simulation of Hopfield network $\mathcal{H}_T$ using simulated annealing (2.2). The input for HTGEN is an object file (in the Wavefront .obj format) describing triangulated surface model T by a list of geometric vertices with their coordinates followed by a list of triangular faces each composed of three vertex reference numbers. The program generates corresponding $\mathcal{H}_T$ which then computes stripification $\Sigma_{\mathbf{y}}$ of T. This is extracted from final stable state $\mathbf{y} = \mathbf{y}^{(\tau^*)} \in Y^*$ of $\mathcal{H}_T$ at microscopic time τ^* into an output .obj file containing a list of tristrips together with vertex data. The user may control the Boltzmann machine by specifying the initial temperature $T^{(0)}$ for (2.2) and the stopping criterion ε given as the maximum percentage of unstable units at the end of stochastic computation.

Program HTGEN has been compared against a leading practical system FTSG that computes stripifications [3]. Apart from other data, experiments have been conducted using "grid" models which are generated by randomly triangulating each square in a $b \times b$ regular grid containing of $n = 2(b-1)^2$ triangles. The average number of tristrips obtained by HTGEN and FTSG are summarized in Table 5.1 where 10 random models were used for each grid size $b = 15, 25, 35$. The results from HTGEN were further averaged for each model over 10 trials of simulated annealing applied for three different initial temperatures $T^{(0)}$ and stopping criteria ε. The corresponding average convergence times τ^* together with the running times in seconds (on common PC) increase as $T^{(0)}$ increases (and ε decreases). Thus $T^{(0)}$ controls the trade-off between the running time and the quality of stripification. One can achieve much better results by HTGEN than by using FTSG with its most successful options (-dfs, -alt) although the running time of HTGEN grows rapidly when the global optimum is being approached. As concerns the time complexity, system HTGEN cannot compete with real-time program FTSG providing the stripifications within a few milliseconds. Nevertheless, HTGEN can be useful if one is interested in the stripification with a small number of tristrips at a preprocessing stage.

Table 5.1: The average number of tristrips for "grid" models obtained by HTGEN and FTSG

n	**HTGEN**			**FTSG**
392	$T^{(0)} = 5$, $\varepsilon = 0.3$ 88 $\tau^* = 23$ (0.10s)	$T^{(0)} = 10$, $\varepsilon = 0.05$ 63 $\tau^* = 166$ (0.72s)	$T^{(0)} = 18$, $\varepsilon = 0.01$ 53 $\tau^* = 1648$ (7.21s)	67
1152	$T^{(0)} = 5$, $\varepsilon = 0.1$ 243 $\tau^* = 442$ (0.76s)	$T^{(0)} = 10$, $\varepsilon = 0.05$ 172 $\tau^* = 347$ (6.01s)	$T^{(0)} = 15$, $\varepsilon = 0.1$ 151 $\tau^* = 1107$ (18.99s)	187
2312	$T^{(0)} = 7$, $\varepsilon = 0.1$ 404 $\tau^* = 117$ (5.31s)	$T^{(0)} = 10$, $\varepsilon = 0.05$ 337 $\tau^* = 489$ (21.29s)	$T^{(0)} = 15$, $\varepsilon = 0.1$ 297 $\tau^* = 1967$ (86.28s)	373

Bibliography

[1] Ackley, D.H., Hinton, G.E., & Sejnowski, T.J. (1985) A learning algorithm for Boltzmann machines. *Cognitive Science* **9**(1):147–169.

[2] Barahona, F. (1982) On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General* **15**(10):3241–3253.

[3] Estowski, R., Mitchell, J.S.B., & Xiang, X. (2002) Optimal decomposition of polygonal models into triangle strips. *Proceedings of the 18th Annual Symposium on Computational Geometry*, 254–263, ACM Press.

[4] Farhat, N.H., Psaltis, D., Prata, A., & Paek, E. (1985) Optical implementation of the Hopfield model. *Applied Optics* **24**(10):1469–1475.

[5] Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA* **79**(8):2554–2558.

[6] Hopfield, J.J. (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA* **81**(10):3088–3092.

[7] Hopfield, J.J., & Tank, D.W. (1985) "Neural" computation of decision in optimization problems. *Biological Cybernetics* **52**(3):141–152.

[8] Pospíšil, D. (2002) Generating triangle strips by Hopfield network. Student's project (in Czech), Faculty of Informatics, Masaryk University, Czech Republic.

[9] Šíma, J., & Orponen, P. (2003) General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation* **15**(12): 2727–2778.