Generating Sequential Triangle Strips by Using Hopfield Nets

Jiří Šíma[1]
Institute of Computer Science, Academy of Sciences of the Czech Republic
P. O. Box 5, 18207 Prague 8, Czech Republic, E-mail: sima@cs.cas.cz

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. The 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 near the global optimum.

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 $m-2$ different triangles $\{v_p, v_{p+1}, v_{p+2}\}$ for $1 \leq p \leq m-2$ so that their shared edges follow alternating left and right turns as indicated in Figure 1 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 [1] which also contains relevant 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 [2].

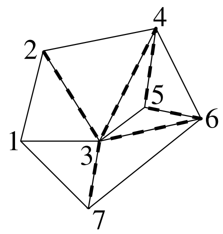

Fig. 1. Tristrip (1,2,3,4,5,6,3,7,1).

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 has formally been verified [3] by proving a one-to-one correspondence between the optimal stripifications of T and the minimum energy states in $\mathcal{H}_T$. Thus, $\mathcal{H}_T$ 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 4. 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.

2 The minimum energy problem

Hopfield networks [4] having well-constrained convergence behavior represent a very influential associative

[1] Research partially supported by projects 1M0021620808, LN00A056 of The Ministry of Education of the Czech Republic.

memory model which is connected to the much-studied Ising spin glass model in statistical physics [5]. Part of the appeal of Hopfield nets also stems from their natural hardware implementations using electrical networks or optical computers, and their application in combinatorial optimization [6].

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.

Also *macroscopic time* $\tau = 0, 1, 2, \ldots$ is introduced during which all the units in the network are updated. A computation of a Hopfield net *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 net is that its dynamics is constrained by *energy* function $E(\mathbf{y}) = -\frac{1}{2} \sum_{j=1}^{s} \sum_{i=1}^{s} w(i,j) y_i y_j + \sum_{j=1}^{s} h(j) y_j$ which is a bounded function defined on its state space whose value decreases along any nonconstant computation path ($\xi_j^{(t)} \neq 0$ is assumed without loss of generality). 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 [4]. Thus the cost function of a hard combinatorial optimization problem can be encoded into the energy function of a Hopfield net 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 [5] (see [7] for related results).

A stochastic variant of Hopfield model called the *Boltzmann machine* [8] 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 $T^{(\tau)} > 0$ is a so-called *temperature* at microscopic time $\tau \geq 0$. This parameter is controlled by *simulated annealing*, e.g. $T^{(\tau)} = T^{(0)}/\log(1+\tau)$ 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 definitions 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 the set of internal edges in triangulation T. Furthermore, a *sequential cycle* is a “cycled tristrip”, that is, an ordered sequence of vertices $C = (v_1, \ldots, v_m)$ where $m \geq 4$ is even, which encodes $m-2$ different triangles $\{v_p, v_{p+1}, v_{p+2}\}$ for $1 \leq p \leq m-2$ so that $v_{m-1} = v_1$ and $v_m = v_2$. 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 2 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 repre-

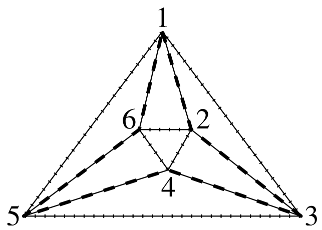

Fig. 2. Sequential cycle (1,2,3,4,5,6,1,2).

sentative 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' 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 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 for each internal edge $e \in I$ define weights $w(i, \ell_e) = -7$ for $i \in J_e \setminus \{r_{e_2}, \ell_e, r_{e_4}\}$ and $w(i, r_e) = -7$ for $i \in J_e \setminus \{\ell_{e_1}, r_e, \ell_{e_3}\}$. 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 as $h(j) = -5 + 2b_{e(j)}$ for $j \in J'$ and $h(j) = 1 + 2b_{e(j)}$ for $j \in J$ 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 [1]. Such infeasible states would have less energy E than those encoding the optimal stripifications [3]. 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 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 energy E when j_C is passive. As depicted in Figure 4 this is implemented by thresholds $h(d_C) = h(a_C) = 1$ and symmetric weights $w(i, d_C) = w(d_C, i) = 2$ for $e(i) \in B'_C$, $w(d_C, j_C) = w(j_C, d_C) = 7$, $w(d_C, a_C) = w(a_C, d_C) = 2$, and $w(j_C, a_C) = w(a_C, j_C) = -2$, 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 $s = 2|I| + 2|\mathcal{C}|$ (similarly the number of connections) 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

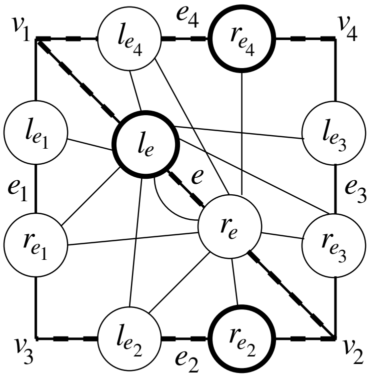

Fig. 3. The construction of $\mathcal{H}_T$ related to $e \in I$.

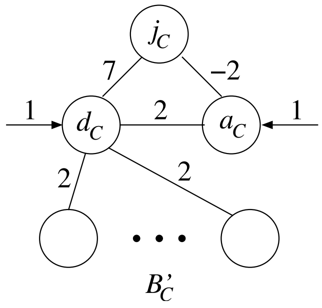

Fig. 4. The construction of $\mathcal{H}_T$ related to $C \in \mathcal{C}$.

Table 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

at most two cycles. In addition, it has been proved [3] that the classes of equivalent optimal stripifications of T are mapped one to one to the minimum energy states that are reached during any sequential computation by $\mathcal{H}_T$ starting at the zero initial state (or $\mathcal{H}_T$ can be initialized arbitrarily if one asymmetric weight is introduced).

4 Experiments

A C++ program HTGEN has been created to automate the reduction from Section 3 including the simulation of Hopfield network $\mathcal{H}_T$ using simulated annealing (see Section 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 of T. This is extracted from final stable state $\mathbf{y}^{(\tau^*)}$ 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 $T^{(0)}$ 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 [1]. 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 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.

References

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

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

[3] Šíma, J. (2004) Tristrips on Hopfield networks. Technical report V-908, Institute of Computer Science, Academy of Sciences of the Czech Republic.

[4] 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.

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

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

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

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