SEARCHING TRIANGLE STRIPS GUIDED BY SIMPLIFICATION CRITERION

O. Belmonte, J. Ribelles, I. Remolar, M. Chover.[1]

Departamento de Informática
Universitat Jaume I, Campus de Riu Sec
12071 Castellón
Spain
{belfern, ribelles, remolar, chover}@uji.es

ABSTRACT

Triangle strips are widely used as a method to accelerate the visualisation process of polygon models in interactive graphics applications. Another widely used method to improve drawing speed is the utilisation of multiresolution models. These models are constructed based on simplification algorithms. None of the current algorithms for searching strips contemplates the posterior simplification of the initial model. In this paper an algorithm for searching strips is presented. The triangles forming a strip are selected based on a simplification criterion according to the average quadratic error associated with the contraction of edges so that the model is simplified. In this manner the strips encountered are conserved as the model is being simplified. The strips generated in this way may be used to draw the polygon model in an incremental form or to transmit it progressively within a computer network.

Keywords: triangle strip searching, interactive visualisation, simplification algorithms, multiresolution models.

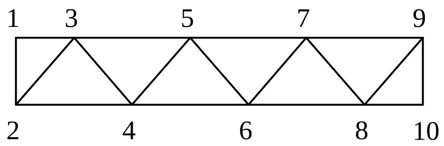
1. INTRODUCTION

In several distinct fields of study such as computer graphics, CAD, digital cartography, medicine, etc. polygon models are among the most widely used to visualise surfaces and volumes. These models routinely may contain a great number of polygons, which directly influences not only the drawing time but also in the time required for their network transmission [Taubi99]. Within the range of polygon models, the most popular is based on triangulation for the simplicity of the algorithms for drawing this graphic primitive, also easy to implement in hardware. To draw independent triangles we need to transmit to the graphic system three vertices for each triangle. However, if in some way we can order the triangles to be drawn, the number of vertices is reduced, reducing the visualisation time as well. An example of such ordering is the *triangle strip* in which the first triangle is specified by three vertices and the following triangles by only one additional vertex. The *triangle strip* primitive is supported within the OpenGL graphics library [Neide99] as well as in Irix-GL. In Fig.1 it is shown a *triangle strip* and the order in which the vertices will be transmitted to the graphic system.

There exist special cases in which the adjacency of the triangles does not permit to directly follow this encoding scheme. In the case shown in Fig.2 once we have drawn v_1 v_2 v_3 v_4 v_5 v_6 v_7 we must break the strip, inverting the order of v_6 and v_7 to continue with the sequence v_7 v_6 v_8 v_9 v_{10} v_{11}. This inversion operation is termed *swap*. The strips for which vertices have been swapped are called generalised triangle strips, if *swaps* are not used the strip is called *sequential*. The inversion instruction is a primitive in the Irix-GL graphics library, but not in OpenGL. In the OpenGL graphic

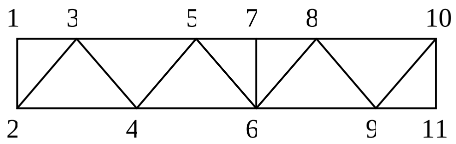
[1] This work is partially supported by Ministerio de Educación y Ciencia CICYT, TIC1999-0510-C02-02

library the swap operation in Fig.2 can be simulated sending again the vertex v_6 after sending vertex v_7. That is, in OpenGL the vertex sequence sent to the graphic system to draw the strip in Fig.2 is v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_6 v_8 v_9 v_{10} v_{11}, which produces a triangle of zero area, called a *degenerate triangle*, formed by the vertices v_6 v_7 v_6.



Triangle strip. The order in which the vertices should be transmitted to the graphic system is v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 and v_{10}

Figure 1



A triangle strip for which we cannot establish a sequence without executing a swap. The order is v_1 v_2 v_3 v_4 v_5 v_6 v_7 swap v_8 v_9 v_{10} v_{11}.

Figure 2

Arkin et al. [Arkin94] demonstrate that to encounter optimal strips in a polyhedral model is NP-complete and so it is necessary to use heuristic strategies in the search for strips. Some of these strategies search a Hamiltonian path in the dual graph. In dual graph of a triangular polyhedral model each node represents a triangle and the arcs represent triangle edges. Each triangle is connected, as a lot, to three neighbours in the case of a manifold surface. A Hamiltonian path traverses a graph visiting each triangle just once. Locating a strip is nothing more than locating a Hamiltonian path in the dual graph, if the path encountered does not contain swaps it is a sequential path.

The algorithm proposed in this paper utilises a spanning tree on the dual graph of the polygon model for the strip search. A spanning tree on a graph is a subset of its arcs in such a way that all the nodes remain connected and the number of arcs is minimum. In this way it is possible to reach any node from the root node.

The principal contributions of this paper are:

- It presents a method for searching triangle strips in a polygon model that are conserved after successive simplifications.
- This algorithm may be the basis for the construction of others which address the problem of incremental visualisation and progressive transmission.

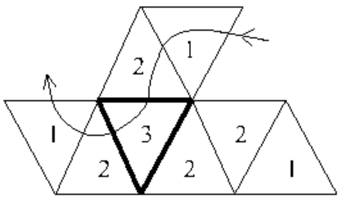
2. PREVIOUS WORK

Previous methods for searching triangle strips fall into two categories: *local* and *global*. The work of Akeley et al. [Akele90], Xiang et al. [Xiang99], Speckmann et al. [Speck97], and Hoppe [Hoppe99] can be classified as *local* because they base their searches in the vicinity of the last triangle in the strip. The work of Santonja et al. [Santo98] can be classified as *global*, because the solution is searched for in a global mode, considering at each new step of the algorithm all triangles of the model. The model of Evans et al. [Evans96] is an algorithm pseudoglobal, the authors naming it so because the initially create regions with the zones of the model that are not triangles, to later triangulate them.

The multiresolution models of Hoppe [Hoppe97] and El-Sana [Elsan99] utilising the triangle strip primitive in the visualisation stage to accelerate the rendering.

2.1 Local algorithms

Historically the first algorithm developed to encounter triangle strips was that of Akeley et al. [Akele90] for Silicon Graphics Inc (SGI). The objective of this algorithm is to find triangle strips in the model in such a way that there remain as few isolated triangles as possible. The algorithm used is a greedy algorithm in that the strips grow according to the local selection of the next triangle as the triangle having the fewest adjacent triangles that have already been allocated to a strip. If two of such triangles exist, the criterion is extended to the neighbours of each candidate. If still the two remain candidates, one is elected at random. In Fig.3 we show an example of the growth of a strip; this one grows in the direction indicated by the arrow, the last triangle introduced into the strip being the one with fatter edges, and each number indicates the number of adjacent triangles.



The growth of a strip utilising the criterion of minimum adjacency, from [Akele90].
Figure 3.

Xiang et al. [Xiang99] utilise an algorithm for the search of strips based on the dynamic programming on a spanning tree generated on a dual graph of the polygon model. This algorithm can also treat models which are not completely triangulated. A first step is to triangulate the model. After that a spanning tree is located on the graph. This tree can be located based on a breadth-first or depth-first search or on a hybrid model over the graph. Finally, the strips are searched on this tree. Each three adjacent triangles form a strip, so the tree is pruned forming a strip with those nodes possessing two children being leaf nodes. The algorithm initially produces strips with few triangles, and therefore afterward the strips are concatenated to form strips as long as possible.

Speckmann et al. [Speck97] developed an algorithm based on a spanning tree to locate triangle strips on triangular irregular networks (TINs) . The principle characteristic of the algorithm is that it does not possess information on adjacencies in graph form, rather it is capable of calculating at run time the antecedent of each triangle according to the criterion of distance of an arbitrary triangle chosen as reference.

Hoppe [Hoppe99] presents an algorithm in which the search criterion is based on the assumption of a cache in the graphic system of a certain size. This cache is the bottleneck in the transmission of vertices to be rendered. Hoppe's algorithm attempts to locate triangle strips in such a way that it can recuperate the most vertices possible of those it holds in cache. The search process simulates the behaviour of the cache to decide which is to be the next triangle to add to the strip.

2.2 Global algorithms

Santonja et al. [Santo98] utilise in the search for strips a genetic algorithm. The objective of this algorithm is to encounter strips that do not contain swaps. Each chromosome representing a possible solution contains as many genes as triangles in the polygon model. The genes inside a chromosome indicate the form in which a triangle is related with its neighbours to form a strip. The function of adaptation is defined according to the criterion that a solution is best when the number of vertices composing the triangle strips is smallest, so that among the model's vertices, triangles, and strips there exists the following relation: V=N+2T where V is the number of vertices which are necessary to transmit to the graphic system, N is the number of triangles in the model and T is the number of strips.

Evans et al. [Evans96] utilise an algorithm which works on models which are not completely triangulated. The objective is the same as in that of Akeley et al. [Akele90]: to leave the smallest amount of isolated triangles. Initially the polygons which are not triangles are grouped in regions to be triangulated at a later moment using one of the three following approximations: i) completely triangulate the model at the beginning of the search; ii) consider only one of the two triangles in cases which they can divide a rectangle when arriving at it; iii) consider the two triangles. With these criteria we decide between multiple options (tie break) to continue the strip in the following manner: i) select as the next triangle the first triangle of those adjacent to the last one introduced in the strip; ii) the same criterion as in Akeley et al. [Akele90]; iii) alternate turns to the left and right; iv) select at random the next one; v) select the option which does not produce swaps and if that is not possible, one at random. The algorithm, called STRIPE, is in the public domain and can be found at the URL http://www.cs.sunysb.edu/~stripe/.

2.3 Triangle strips in multiresolution models

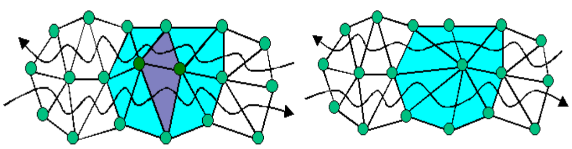
Hoppe [Hoppe97] has utilised the triangle strips in the render stage inside a view dependent multiresolution model. After selecting which triangles to draw, triangle strips are searched on them. Throughout experimentation Hoppe concludes that the fastest triangle strips search algorithm is a greedy one. In this greedy algorithm each of the non drawn triangle begins a new strip growing throw its non-rendered neighbours. In order to reduce the strip fragmentation the strips are grown in a spiral clockwise manner.

El-Sana et al. [Elsan99] have developed a view dependent multiresolution model based on an edge collapsing criterion. The first step in constructing the model is to search triangle strips on it. These triangle strips are stored in a data structure called skip list [Pugh90]. Once the multiresolution model has determined which strips visualise, the skip list is processed. If none of its triangles has been collapsed the strip is drawn, otherwise the skip list is processed in order to eliminate the collapsed triangles.

In none of the summarised work, the simplification criterion guides the strip search. In [Hoppe97] the search criterion is to avoid the strip fragmentation and this search is made frame by frame, no relation exists between strips in contiguous frames. In [Elsan99] the initial strips are maintained between frames but the criterion to search them is not based on simplification so these strips will be break when they will be simplified.

3. CONTRIBUTIONS OF THE WORK

None of the algorithms summarised in the above section has as their final objective anything other than to search strips as long as possible. Once the strips have been encountered they serve to accelerate the drawing process. None of the algorithm summarised exploits the conservation of strips for an incremental rendering or for the progressive transmission through computer networks. Progressive transmission is a technique in which the polygon model is sent first in a simplified version of the original model, transmitting later on the information necessary to refine the model until finally reaching the detail of the original.



A strip and its result upon contraction of an edge of the border between strips.
Figure 4.

Taking this into account the primary objective of the algorithm proposed in this paper is to find triangle strips in the polygon model serving as a base for drawing the model in a incremental manner or to transmit it progressively. Obviously, the model also will be drawn faster using triangle strips.

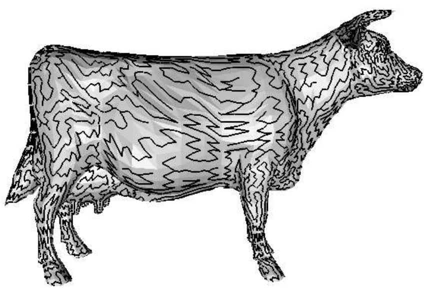
The algorithm initially constructs a spanning tree on the dual graph of the model. The creation of this tree is made based on the simplification criterion of Garland and Heckbert [Garla97]. Finally, the tree is traversed to encounter the triangle strips. The strips encountered in this way have the following property: because the strips have been created by a simplification criterion, when simplifying the original model they tend to conserve themselves, that is, the simplifications will fragment as little as possible the strips of the original model and it do not need to send two extra vertices at the start of the strip to the graphic system (see Fig.4).

4. SPANNING TREES, CRITERION FOR EDGE COLLAPSING, AND HOW TO JOIN THEM

4.1 Spanning trees

As it was said in the introduction, a spanning tree on a graph is a subset of its arcs in such a way that all the nodes remain connected and the number of arcs is minimum. If this graph is the dual graph of the model a polygon model, traversing the tree it will visit all the triangles forming the model (see Fig.5).

To create the spanning tree over the polygon model the method uses the algorithm of Prim [Brass97]. This algorithm works on weighted graphs. This algorithm finds a minimum spanning tree over the original graph. The algorithm guarantees that the sum of the arcs forming the tree is minimum, no other combination of arcs forming a tree has a smaller total weight.



A polygon model with its spanning tree.
Figure 5.

In this algorithm a candidate is selected at each moment as the best one according to certain criterion. This candidate edge will pertain to the solution set if one of its nodes already pertains to the tree and the other not, rejecting it in any other case. To accelerate the tree construction process a list of priorities with the edges that form the border of the tree is maintained.

The selection of the best candidate edge is where we find the novelty of the algorithm. Because the final objective is that the strips encountered are conserved as the model is simplified, we should select as candidate that edge which has the least probability of being eliminated in the simplification process, or stated in another way, we should select the edge that possesses a higher error, because it will be eliminated in the final simplification process.

4.2 Criterion for edge collapsing

The criterion that we have used to select the edges of the solution set is based on the mean error of the simplification criterion of Garland and Heckbert [Garla97]. This method is based on edge contraction. Each edge contraction has associated an error calculated as $\Delta(v) = v^TQv$, where Q is a symmetric 4x4 matrix associated with each vertex. On the other hand the error of the vertex with respect to the set of planes that share it is defined as the sum of squares distances to these planes:

$$\Delta(v) = \Delta\left(\left[v_x, v_y, v_z\right]^T\right) = \sum_{p \in planes(v)} \left(p^T v\right)^2 \qquad (1)$$

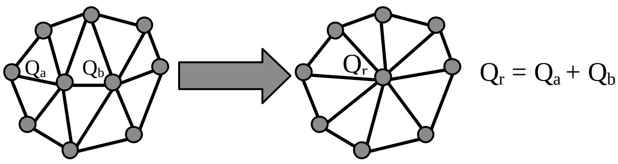
where $p=[A\ B\ C\ D]^T$ represents the plane defined by the equation $Ax+By+Cz+D=0$. The normal vector of the plane is normalised so $A^2+B^2+C^2=1$. (1) can be written as:

$$\Delta(v) = \sum_{p \in planes} \left(v^T p\right)\left(p^T v\right) = v^T \left[\sum_{p \in planes} pp^T\right] v \qquad (2)$$

and we can identify

$$Q = \sum_{p \in planes} pp^T \qquad (3)$$

For an edge collapsing (v1, v2) $\rightarrow$ v, v is chosen in such a way that $\Delta(v) = v^TQv$ will be minimum. The error matrix Q associated with the new vertex is calculated as $Q_r = Q_a+Q_b$. The value $\Delta(v)$ represents the cost of the edge contraction as it was said, see Fig.6.



Contraction of an edge during the simplification process. The error associated with the vertex that remains is the sum of the errors of the original vertices.
Figure 6.

4.3 The join

The spanning tree is constructed using the Prim's algorithm. There are two triangles in the polygon model sharing the maximum error edge. The root of the tree will be that having the maximum error edge of the four remaining edges.

As next candidate we take the head of the list of priorities with the edges, if one of the triangles sharing this edge already pertain to the tree and the other not, it will be added to the solution set.

Since the method takes the edge with maximum weight it could be said that it finds a maximum spanning tree over the polygon model.

To locate this new triangle in the spanning tree the following criterion is used: the new triangle will be set as left child if the triangle already pertaining to the solution set has no children, it will be set as right child if it do not. This criterion will do the triangle strips easier to search in the next stage.

Once all the triangles in the model have been added to the tree, the algorithm proceeds to search the triangle strips. This begins at the root node, the next triangle in the strip will be that at the left child if is no visited, and the right node if it is.

This criterion will be break if the current triangle strip arrives to one triangle with all of its neighbours visited. This triangle will be the last one in this strip and it is called an *isolated triangle*.

5. RESULTS AND CONCLUSIONS

The main objective of the algorithm presented in this paper is to encounter triangle strips that are conserved as the model is simplified. In such a manner that the appearance of the object is represented by the strips. Fig.7 in appendix A shows an image sequence of the original model, five levels of detail and the strips after simplification. Observing carefully this sequence it can be appreciated how the strips are conserved between images. This is due to the fact that the edges which are initially contracted are at strip borders. The sequence of images in Fig.8 shows the details of a head at maximum level of detail and five other resolutions, showing again the original strips and those remaining after successive simplifications.

We compare our extraction algorithm with the Stripe algorithm in the following way: we count the number of strips in the model as this is simplified. To compare the two algorithms the number of strips is normalised to the number of strips in the initial model. In appendix B Fig.9 (a) and (c) show the number of strips in the bunny and cow model as these are simplified. Fig.9 (b) and (d) show the variation of the number of strips (derivative) normalised to the number of strips in the original model as these are simplified.

It is quit clear to observe from Fig.9 (b) and (d) that the variation in the number of strips as the model is simplified is proportionally less in our method that in the Stripe method.

On the other hand, the number of vertices sent to the graphic systems is about 45% minor than the number of vertices sent like independent triangles. With the Stripe method this percentage grows until the 60%.

6. SUMMARY AND FUTURE WORK

We presented a method for searching triangle strips in polygon models with the property that the strips produced are conserved after successive simplifications of the model. This algorithm may be the basis for the construction of others which address the problem of incremental visualisation and progressive transmission. The number of strips encountered is greater than those encountered in the popular Sripe algorithm, but this is not a point against it for two reasons:

- Assuming that the strips contain no swaps, the savings in number of vertices transmitted to the graphic system is similar in the two algorithms.
- Hoppe [Hoppe99] demonstrates that large strips are not always the most efficient for visualisation;

A possible direction for future work is to guide the search not only with the edge collapse error information in the original model but in all of the representations as the model is simplified.

REFERENCES

[Akele90] K. Akeley, P. Haeberli, and D. Burns. tomesh.c: Developer's Toolbox CD, 1990.

[Arkin94] E. Arkin, M. Held, J. Mitchell and S. Skiena. "Hamiltonian triangulatios for fast rendering". Second Annual European Symposium on Algoritms, Springer – Verlag Lecture Notes in Computer Science. Vol: 855, pages: 36 – 47. 1994.

[Brass97] G. Brassard, and P. Bratley. "Fundamentos de algoritmia". Prentice Hall. 1997.

[Elsan99] El-Sana, Jihad et al, "Skip Strips: Maintaining Triangle Strips for View-dependent Rendering", IEEE Visualisation '99, pp.131–138, 1999.

[Evans96] F. Evans, S. S. Skiena, and A. Varshney. "Optimizing Triangle Strips for Fast Rendering", IEEE Visualisation '96, pp.319-326, 1996.

[Garla97] M. Garland, and P. Heckbert, "Surface Simplification Using Quadric Error Metrics", SIGGRAPH '97, pp. 209-216, 1997.

[Hoppe97] H. Hoppe. "View-dependent refinement of progressive meshes", SIGGRAPH '97, pp.189-197, 1997.

[Hoppe99] H. Hoppe. "Optimization of mesh locality for transparent vertex caching". SIGGRAPH '99, pp.269 – 276, 1999.

[Luebk97] D. Luebke and C. Erikson. "View-Dependent Simplification of Arbitrary Polygonal Environments", SIGGRAPH '97, pp.199-208, 1997.

[Neide93] J. Neider, T. Davis and M. Woo. "OpenGL Programming Guide", Addison – Wesley, 1993.

[Pugh90] W. Pugh. "Skip lists: A probabilistic alternative to balanced trees", Communications of the ACM, Vol 33(6), pp. 668-678, 1990.

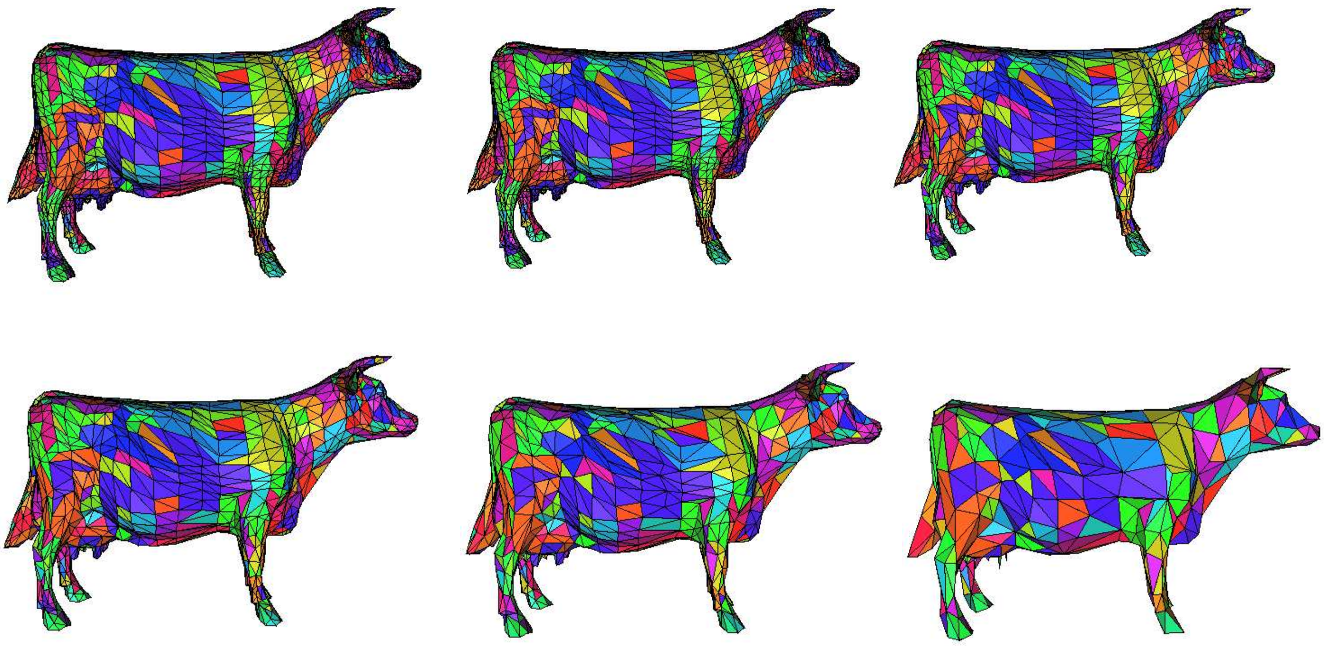
[Santo98] J. Santonja and S. Bayarri, "Resolución del problema de extracción de tiras triangulares mediante algoritmos genéticos", Actas del VII Congreso Español de Informática gráfica, pp.149-162, 1998.

[Speck97] B. Speckmann and J. Snoeyink, "Easy triangle strips for TIN terrain models", 9th Canada Conference on Computational Geometry, pp.239-244, 1997.

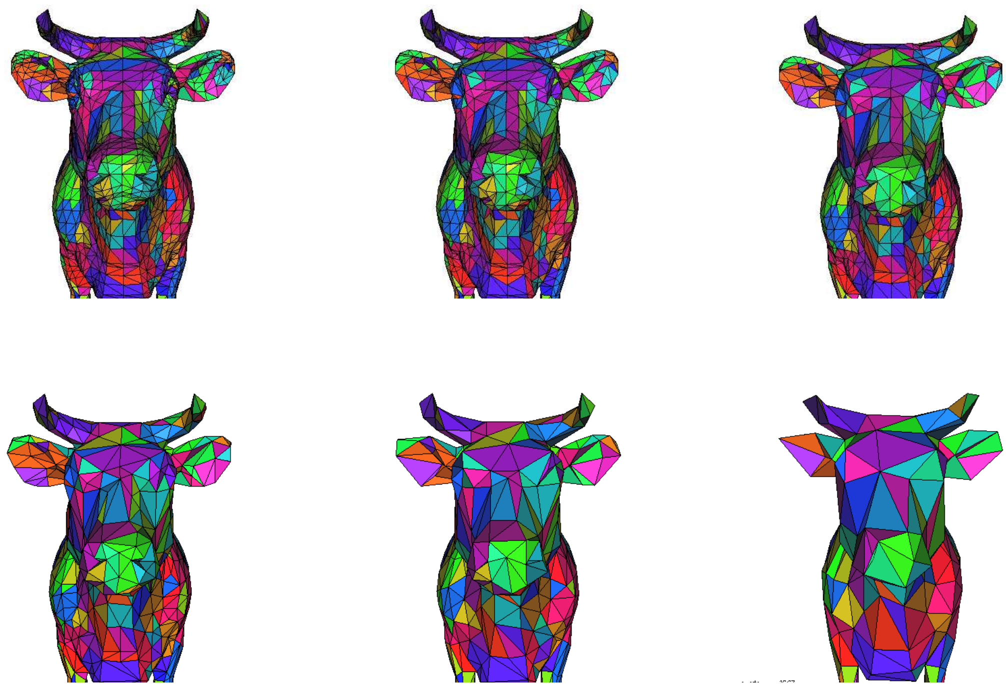
[Taubi99] G. Taubin. "Geometric Compression and Progressive Transmission". EUROGRAPHICS '99, pp.81-96, 1999.

[Xiang99] X. Xiang, M. Held and J.S.B. Mitchell. "Fast and Effective Stripification of Polygonal Surface odels". Symposium on Intereactive 3D Graphics, pp.71-78, 1999.

APPENDIX A



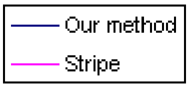
Strips found in a polygonal model and the results after contracting 500, 1000, 1500, 2000 and 2500 edges.
Figure 7.

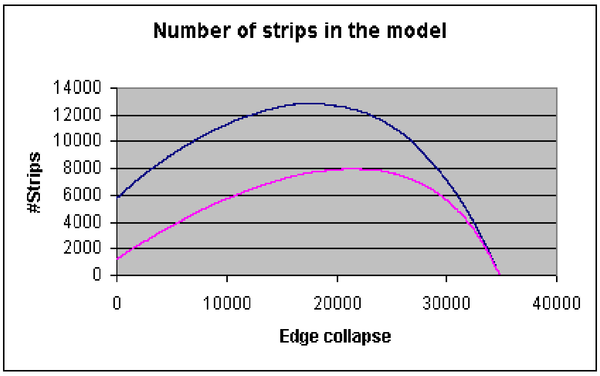


Details of the head; original model and the results after contracting 500, 1000, 1500, 2000 and 2500 edges.
Figure 8.

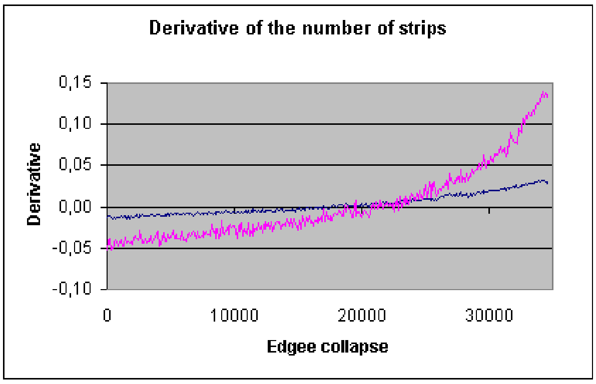
APPENDIX B

	#Strips	#Vertices	#Triangles
Our method	5,785	34,834	69,451
Stripe	1,229		

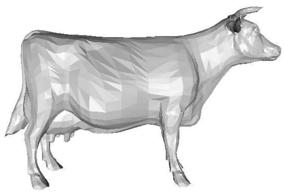




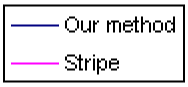
(a) Number of strips in the model as it is simplified

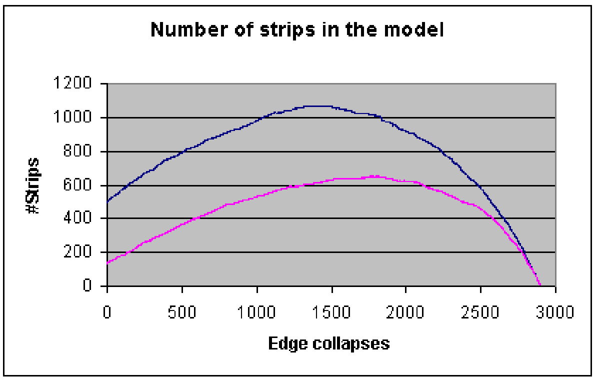


(b) Normalised variation in the number of strips as the model is simplified

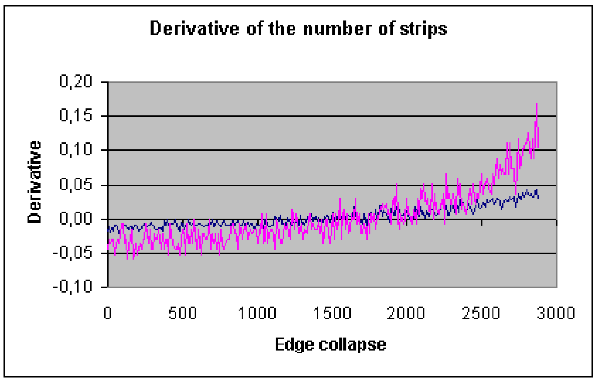


	#Strips	#Vertices	#Triangles
Our method	535	2,904	5,804
Stripe	155		





(c) Number of strips in the model as it is simplified



(d) Normalised variation in the number of strips as the model is simplified

Results obtained for the cow and bunny models.
Figure 9.