Búsqueda de tiras de triángulos guiada por un criterio de simplificación*

O. Belmonte, J. Ribelles, I. Remolar, M. Chover.

Departamento de Informática. Universitat Jaume I.
Campus de Riu Sec, 12071 Castellón
Tlf: 964.728.316 Fax: 964.728.435. e-mail: belfern@inf.uji.es

Resumen

Un método de aceleración ampliamente utilizado para la visualización de modelos poligonales basados en triángulos es el uso de la primitiva tira de triángulos. Otro método ampliamente utilizado para aumentar la velocidad de dibujado en aplicaciones gráficas interactivas es la utilizaciones de modelos multirresolución, construidos a partir de algoritmos de simplificación. Ninguno de los algoritmos actuales de búsqueda de tiras tiene en cuenta la posterior simplificación del modelo inicial. Presentamos un algoritmo de búsqueda de tiras en el que el criterio de selección de los triángulos que forman las tiras está basado en la medida del error cuadrático asociado a la contracción de las aristas a medida que el modelo se simplifica; de este modo las tiras encontradas se conservan a medida que el modelo original se va simplificando. Las tiras así generadas pueden utilizarse para dibujar el modelo de forma incremental o para transmitirlo progresivamente por un red de ordenadores.

1. Introducción

En distintos campos de la informática como informática gráfica, CAD, cartografía, medicina, etc, los modelos poligonales son los más extendidos para visualizar superficies y volúmenes. Estos modelos actualmente pueden llegar a contener un gran número de polígonos, lo que influye directamente tanto en el

* Trabajo financiado por el proyecto TIC1999-0510-C02-02 (CICYT, Ministerio de Educación y Ciencia)

tiempo de dibujado como en el tiempo empleado en su transmisión por una red de ordenadores [1]. Dentro de los modelos poligonales el más popular es el basado en triángulos por la sencillez de los algoritmos de dibujado de esta primitiva gráfica, fáciles de implementar en hardware. Para dibujar los triángulos independientemente necesitamos transmitir al procesador gráfico tres vértices por cada triángulo, pero, si de algún modo ordenamos los triángulos a dibujar, este número de vértices disminuye, disminuyendo asimismo el tiempo de visualización. Una ordenación de este tipo es la tira de triángulos en la que el primer triángulo de la tira queda especificado por los tres primeros vértices y los siguientes triángulos solo por un vértice adicional. Esta primitiva gráfica es soportada tanto por la librería gráfica OpenGL [2] como por la Irix-GL, En la figura 1 se muestra una tira y el orden en el que hay que transmitir los vértices al procesador gráfico.

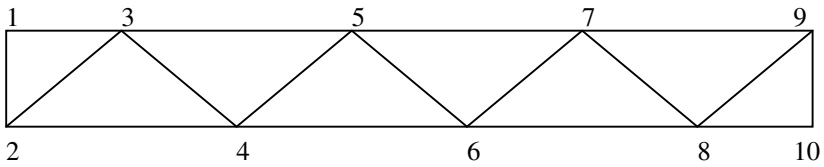

Figura 1: Una tira de triángulos. El orden en que los vértices deben ser transmitidos al procesador gráfico es: 1 2 3 4 5 6 7 8 9 y 10.

Existen casos en los que la adyacencia de los triángulos no permite utilizar directamente esta codificación, en el caso de la tira de triángulos de la figura 2 una vez dibujada la secuencia 1 2 3 4 5 6 7 hemos de romperla invirtiendo el orden de los vértices 6 y 7 para continuar con la secuencia 7 6 8 9 10 y 11, a esta operación se le llama 'swap'. A las tiras de triángulos que incorporan el intercambio de vértices se les llama tiras de triángulos generalizadas. La instrucción de inversión es una primitiva en la librería gráfica Irix-GL, pero no en OpenGL, por lo que hay que simularla enviando el penúltimo vértice antes de seguir dibujando, es decir, en OpenGL la secuencia de vértices a transmitir al procesador gráfico para dibujar la tira de la figura 2 es: 1 2 3 4 5 6 7 6 8 9 10 y 11, lo que produce un triángulo de área 0, al que llamaremos degenerado, formado por la secuencia de vértices 6 7 6.

Arkin et al. [3] demuestran que encontrar tiras óptimas en un modelo poliédrico es NP-completo por lo que se hace necesaria la utilización de estrategias heurísticas en la búsqueda de las tiras. Algunas de estas estrategias buscan un camino Hamiltoniano en el grafo dual. El grafo dual de un modelo poliédrico basado en triángulos es aquel en el que cada nodo del grafo representa un triángulo y los arcos del grafo representan las aristas del triángulo que le conectan a sus tres

posibles vecinos en el caso de superficies manifold. Encontrar una tira no es más que encontrar un camino Hamiltoniano en el grafo dual, si el camino encontrado no contiene swaps se dice que el camino es secuencial. El algoritmo propuesto utiliza, en la búsqueda de las tiras, un árbol de recubrimiento sobre el grafo dual del modelo.

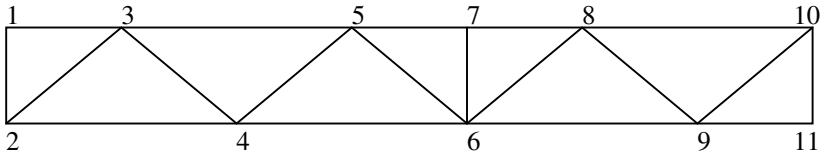

Figura 2: Una tira de triángulos en el que no se puede establecer una secuencia sin emplear un 'swap'. El orden es: 1 2 3 4 5 6 7 swap 8 9 10 y 11.

2. Trabajo previo

Históricamente el primer algoritmo desarrollado para encontrar tiras de triángulos es el de K. Akeley et al. [4] para SGI. El objetivo de este algoritmo es encontrar tiras de triángulos en el modelo de tal modo que queden el menor número posible de triángulos aislados. El algoritmo utilizado es un algoritmo voraz en el que las tiras van creciendo a través de la selección local del siguiente triángulo como aquel que tiene el menor número de triángulos adyacentes que ya han sido incorporados a alguna tira. Si existen dos, el criterio se extiende a los vecinos de cada uno de los posibles candidatos. Si aún así las dos posibilidades siguen siendo válidas, se elige uno de ellos al azar. En la figura 3 se muestra un ejemplo de crecimiento de una tira; esta crece en la dirección indicada por la flecha, el último triángulo introducido en la tira es el de aristas gruesas, cada número indica el número de triángulos adyacentes.

F. Evans et al. [5] utilizan un algoritmo que trabaja sobre modelos que no están completamente triangulados. El objetivo del algoritmo es el mismo que el de Akeley et al. [4], dejar el menor número posible de triángulos aislados. El modo de proceder es el mismo que el de Akeley et al. [4], introducir como siguiente triángulo en la tira aquel que contenga el menor número de triángulos que ya han sido incorporados a alguna. Inicialmente los polígonos que no son triángulos son agrupados en regiones para más tarde triangular el modelo utilizando una de las

tres aproximaciones siguientes i) triangular completamente el modelo al inicio de la búsqueda; ii) considerar únicamente uno de los dos triángulos en los que se puede dividir un rectángulo al llegar a él; iii) considerar los dos triángulos. Con estos criterios se decide entre las múltiples opciones (tie break) para continuar la tira del siguiente modo i) elegir como siguiente el primer triángulo de los adyacentes al último introducido en la tira; ii) el mismo criterio que Akeley et al. [4]; iii) alternar giros a izquierda y derecha; iv) elegir aleatoriamente el siguiente; v) elegir la que no produce swap y si no es posible, uno al azar. El algoritmo, llamado STRIPE, es de dominio público y se puede encontrar en la dirección Internet http://www.cs.sunysb.edu/~stripe/

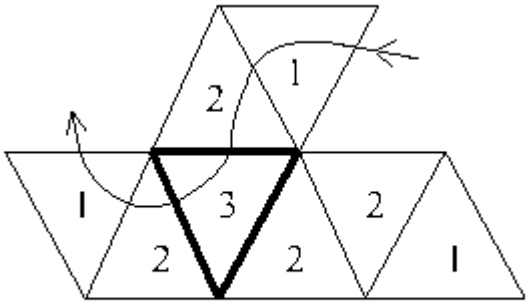

Figura 3: Crecimiento de una tira utilizando el criterio de mínima adyacencia de [4].

X. Xiang et al. [6] utilizan en la búsqueda de tiras un algoritmo de programación dinámica sobre un árbol de recubrimiento generado sobre el grafo dual del modelo poligonal. Este algoritmo también puede tratar modelos que no están completamente triangulados. Un primer paso es triangularlo y posteriormente encontrar un árbol de recubrimiento sobre el grafo que puede estar basado en búsqueda en anchura, profundidad o en un modelo híbrido. Sobre este árbol se buscan las tiras, para ello, y ya que tres triángulos adyacentes siempre forman una tira, se poda el árbol formando una tira con todo aquel nodo que posea dos hijos ambos nodos hoja. El algoritmo produce inicialmente tiras con pocos triángulos, por lo que después se concatenan las tiras según sea posible.

Santonja et al. [7] utilizan en la búsqueda de las tiras un algoritmo genético. El objetivo del algoritmo es encontrar tiras que no contenga swaps, para ello cada uno de los cromosomas de su algoritmo genético contiene tantos genes

como triángulos contenga el objeto y cada gen indica la forma en que un triángulo se relaciona con los demás para formar tiras. La función de adaptación se define según el criterio de que una solución es tanto mejor cuanto menor sea el número de vértices que compongan las tiras triangulares que propone, ya que entre los vértices, triángulos y tiras de un modelo existe la siguiente relación: V = N+2T donde V es el número de vértices que es necesario transmitir al procesador gráfico, N el número de triángulos del modelo y T el número de tiras.

Speckmann et al. [8] han ideado un algoritmo basado en árbol de recubrimiento para encontrar tiras de triángulos sobre TINs (triangulated irregular network). La principal característica de este algoritmo es que no posee información de adyacencia en forma de grafo sino que es capaz de calcular en tiempo de ejecución el antecesor de todo triángulo según un criterio de distancia a uno arbitrariamente elegido como referencia.

Hoppe [9] presenta un algoritmo en el que el criterio de búsqueda está basado en el supuesto de la existencia de una caché en el procesador gráfico de cierto tamaño, y ya que el cuello de botella es la transmisión de los vértices a dibujar, intenta encontrar tiras de triángulos de tal modo que pueda recuperar cuantos más vértices mejor de los que actualmente tiene en la caché. El proceso de búsqueda simula el comportamiento de la caché para decidir cuál es el siguiente triángulo a añadir a la tira.

Los criterios para buscar tiras de triángulos en los trabajos previos van desde dejar el menor número de triángulos aislados como los de Akeley ete al. [3] Evans et al. [5], encontrar tiras cuanto más largas mejor Xiang et al. [6], evitar swaps Santonja et al. [7] o simular el comportamiento del procesador gráfico Hoppe [9]. Estos criterios se pueden clasificar como locales: Akeley et al. [3], Xiang et al. [6], Speckmann et al [8] y Hoppe [9] porque basan su búsqueda en la vecindad del último triángulo introducido en la tira; o globales: Santonja et al. [7], en el que la solución se busca de modo global, considerando en cada nuevo paso del algoritmo todos los triángulos del modelo. El de Evans et al [5] es un algoritmo pseudo - global, los autores lo llaman global porque inicialmente crean regiones con las zonas del modelo que no son triángulos, para más tarde irlas triangulando.

Hoppe [13] ha utilizado tiras de triángulos dentro de un modelo mutiresolución dependiente de la vista en la etapa de dibujado. Tras determinar los triángulos a dibujar en un cierto instante, se buscan sobre ellos tiras de triángulos para acelerar el proceso de dibujado. Mediante la experimentación llega a la conclusión de que el método más rápido de búsqueda de tiras está basado en un algoritmo voraz: cada triángulo no dibujado inicia una tira, la tira crece hacia los

vecinos que aún no han sido dibujados. Para reducir la fragmentación de las tiras el algoritmo hace crecer estas en espirales en el sentido de las agujas del reloj.

La idea presentada por El-Sana et al [14] para su modelo multirresolución dependiente de la vista es opuesta al modelo de Hoppe. Este modelo multirresolución utiliza como algoritmo de simplificación cualquiera que esté basado en la contracción de aristas. El-Sana et al. buscan las tiras en el modelo original basándose en el algoritmo de [5]. Las tiras así encontradas se representan en una estructura llamada skip lists [15]. Una vez que su modelo multirresolución ha determinado qué triángulos dibujar se comprueba cada una de las tiras originales; si la tira no posee ningún triángulo con alguna arista colapsada, se dibuja tal cual, de lo contrario se ha de procesar la tira para eliminar los triángulos con aristas colapsadas.

En ninguno de los anteriores trabajos la búsqueda de tiras es uno de los pilares sobre los que se soporta el modelo multirresolución. En [13] el criterio de crecimiento de tiras es que estas se fragmenten lo menos posible, pero esta búsqueda se hace frame a frame, no existiendo ninguna relación entre las tiras encontradas en un frame y en el siguiente. El-Sana et al [14] conservan las tiras iniciales y las procesan en cada frame antes de dibujarlas, pero el algoritmo para encontrarlas, el de [5], no tiene en cuenta en ningún momento el criterio de simplificación utilizado en la generación del modelo multirresolución.

3. Contribución del trabajo

Ninguno de los algoritmos revisados en el apartado anterior tiene como objetivo final otro que no sea el crecimiento de tiras. Una vez encontradas estas tiras sirven para acelerar el proceso de dibujado, pero ninguno de ellos explota la conservación de las tiras para un dibujado incremental o para la transmisión progresiva por una red de ordenadores, técnica en la que un modelo poligonal es enviado primero como una versión simplificada del original transmitiendo a continuación información para refinarla hasta alcanzar el detalle del modelo original.

Teniendo esto en cuenta, el objetivo principal del algoritmo propuesto para encontrar tiras de triángulos en modelos poligonales es que éstas, además de acelerar el proceso de dibujado, también posean la propiedad de ser base para poder dibujar el modelo de forma incremental o transmitirlo de modo progresivo.

El algoritmo inicialmente construye un árbol de recubrimiento sobre el grafo dual del modelo. La creación de este árbol se hace basándose en un criterio de simplificación. Finalmente el árbol es recorrido para encontrar las tiras de triángulos. Las tiras así encontradas cuentan con la siguiente propiedad: ya que la creación de las tiras ha sido guiada por un criterio de simplificación, al simplificar el modelo original, éstas tienden a conservarse, es decir, las simplificaciones fragmentarán lo menos posible las tiras del modelo original y no tendremos que mandar dos vértices extra de inicio de tira al procesador gráfico.

4. Árboles de recubrimiento, criterios de colapse de arista y como unirlos.

Un árbol de recubrimiento sobre un grafo es un subconjunto de sus aristas de tal modo que todos los nodos quedan conectados; si es el grafo dual del modelo, recorriendo el árbol visitaremos todos los triángulos que lo forman.

Para crear el árbol de recubrimiento utilizamos el algoritmo de Prim [10], seleccionando en cada momento como arista candidata la mejor, bajo cierto criterio que expondremos más tarde, de las que aún no pertenecen al conjunto solución y pertenecerá a la solución si uno de los nodos ya pertenece al árbol y el otro no, rechazándola en cualquier otro caso. Para acelerar el proceso de construcción del árbol mantenemos una lista de prioridad con las aristas que forman la frontera del árbol que estamos construyendo ya que el algoritmo de Prim rechaza la arista candidata si no conecta por uno y solo uno de sus nodos con el conjunto solución.

En la selección de 'la mejor' arista es donde radica la novedad del algoritmo. Ya que el objetivo último es que las tiras encontradas se conserven según el modelo se vaya simplificando, debemos seleccionar como candidata aquella arista que tenga una menor probabilidad de ser eliminada en el proceso de simplificación, o dicho de otro modo, debemos seleccionar aquella que, posea un mayor error , ya que será eliminada al final del proceso de simplificación.

Al utilizar un árbol de recubrimiento en la búsqueda de las tiras de triángulos garantizamos que el algoritmo considera al objeto como un todo, es decir, el algoritmo efectúa una búsqueda global de las tiras de triángulos en el modelo.

El criterio que hemos utilizado para seleccionar las aristas del subconjunto solución está basado en la medida de error del criterio de simplificación de Garland et al. [11]. Este método de simplificación está basado en la contracción de aristas. En él cada vértice tiene asociada una matriz de error **Q** calculada a partir de las normales de los triángulos que comparten dicho vértice. El error es la distancia mínima del vértice a cada uno de los triángulos a los que pertenece el vértice. Siendo **v** el vector de posición del vértice este error se calcula como $\mathbf{v}^T\mathbf{Q}\mathbf{v}$, que inicialmente es cero para cada uno de los vértices del modelo. Una vez calculado este error para cada una de las aristas del modelo podemos ordenarlas de mayor a menor error. Cada contracción elimina uno de los dos vértices de cada arista y asocia al vértice que no se elimina, como nueva matriz de error, la suma de las matrices de los dos vértices iniciales. El error asociado la contracción de la arista se calcula multiplicando la matriz de error resultado por el vector de posición del vértice que se mantiene tras la contracción. En la figura 4 se muestra un ejemplo de la contracción de una arista.

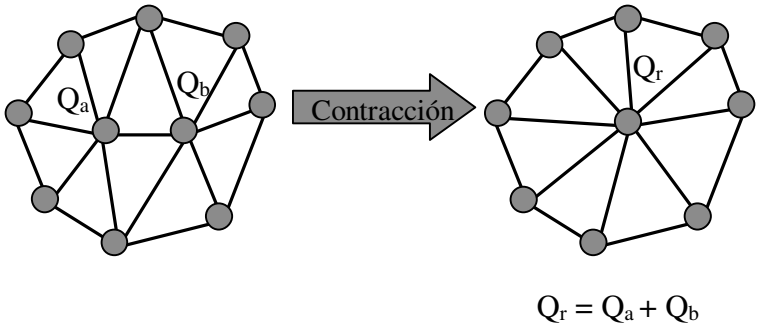

Figura 4: Contracción de una arista en el proceso de simplificación. El error asociado al vértice que permanece es la suma de los errores de los vértices originales.

Con este criterio de selección de la siguiente arista candidata y sobre el grafo dual del modelo construimos el árbol de recubrimiento sobre el modelo poligonal. El nodo raíz del árbol de recubrimiento será aquel triángulo que posea la arista, de la totalidad del modelo original, con mayor error tras una contracción. Si al añadir un nodo, que representa un triángulo en el modelo, el nodo al que está conectado por la arista candidata no posee ningún hijo, lo colocamos como hijo a su izquierda, en caso contrario lo colocamos como hijo a la derecha. De este modo, la búsqueda de tiras la haremos visitando de cada nodo, primero el hijo situado a la

izquierda, el de mayor error de los dos hijos, y si este camino ya ha sido explorado, el hijo situado a la derecha. Si un nodo no posee alguno de sus hijos y de poseer alguno este ya ha sido visitado, saltamos dentro del árbol al nodo que esté unido con el actual por la arista con mayor error y no haya sido aún visitado.

Si hacemos crecer las tiras atravesando las aristas con mayor error, al ser estas las últimas que se contraen cuando el modelo se va simplificando, las tiras no quedarán cortadas, ya que las aristas que se contraerán serán aquellas que sean frontera entre dos tiras adyacentes. En la figura 5 podemos ver un ejemplo de ello.

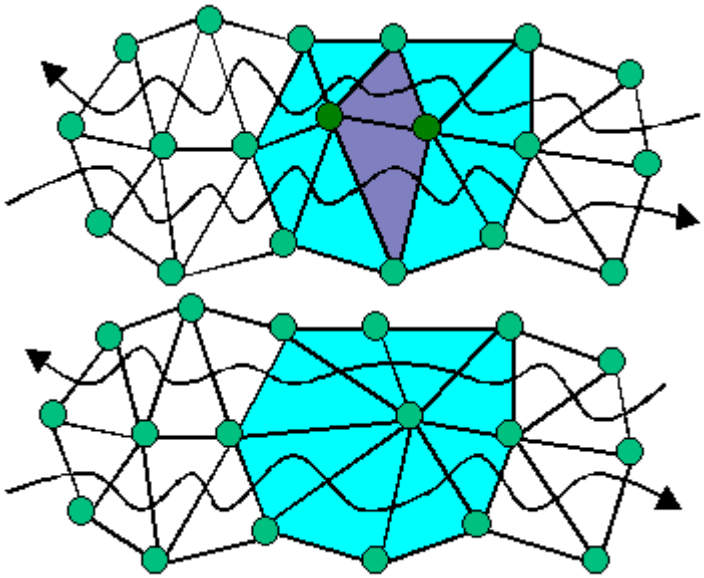

Figura 5: Una tira y la resultante al contraer una arista de la frontera entre tiras.

Para eliminar en lo posible la existencia de triángulos aislados, el algoritmo de recorrido del árbol antes de seleccionar la arista de mayor error inspecciona si, de entre los vecinos del triángulo actual, queda algún vecino cuyos vecinos a su vez han sido todos visitados. De encontrarse alguno, se toma como siguiente, y último, en la tira.

5. Resultados

Recordemos que el objetivo último de este algoritmo es encontrar tiras de triángulos que se conserven, que no se corten, a medida que el modelo se simplifique, de tal forma que la apariencia del objeto de algún modo esté representada por estas tiras. El experimento diseñado para comprobar que las tiras se conservan al ir simplificando el modelo compara el modelo original con una secuencia de simplificaciones utilizando el método de Garland et al. [11]. Para hacer el experimento más sencillo, se ha generado un modelo multirresolución [12] sobre el modelo poligonal original, y de este hemos extraído diferentes niveles de resolución.

En la figura 6 se puede ver una secuencia de imágenes mostrando el modelo original y cinco niveles de detalle mostrando las tiras tras la simplificación. Al observar detenidamente esta secuencia se puede apreciar como las tiras se conservan entre imágenes, esto como ya sabemos es debido a que las aristas que inicialmente se contraen están localizadas en la frontera entre tiras. La secuencia de imágenes de la figura 7 muestra un detalle de la cabeza a máxima resolución y cinco resoluciones, mostrando de nuevo las tiras originales y las que permanecen tras sucesivas simplificaciones.

Sobre estas figuras queremos hacer dos observaciones:

I) Aproximadamente el 45% de las aristas que se contraen pertenecen a la frontera entre tiras.

II) A medida que contraemos más aristas la proporción de tiras cortadas aumenta.

6. Conclusiones y trabajo futuro

Se ha presentado un método de búsqueda de tiras en modelos poligonales con la propiedad de que las tiras halladas se conservan tras sucesivas simplificaciones del modelo. Este algoritmo puede ser la base en la construcción de otros que aborden el problema de la visualización incremental y la transmisión progresiva. El número de tiras encontradas es aproximadamente cinco veces mayor de las que encuentra el popular algoritmo STRIPE, pero ello no es ningún punto en contra por tres motivos: i) Atendiendo a la relación ya mencionada entre el número de vértices que se le deben pasar al procesador, el número de triángulos del modelo y el número de tiras encontradas y suponiendo que en las tiras no contiene ‘swaps’,

el ahorro en el número de vértices transmitidos al procesador si el modelo se dibuja como triángulos independientes frente al dibujado con tiras es similar en los dos algoritmos; ii) Hoppe [9] demuestra que no siempre las tiras más largas son las más eficientes en visualización; iii) El objetivo del algoritmo propuesto es la conservación de las tiras conforme el modelo se simplifica, no extraer el menor número de tiras posible.

Una posibles dirección de trabajo futuro es la construcción de un modelo multirresolución basado en tiras de triángulos e integrar este modelo como un nodo PROTOTYPE en el estándar VRML.

Referencias.

[1] G. Taubin. "·D Geometric Compression and Progressive Transmission". EUROGRAPHICS '99, STAR pag: 81-96. Septiembre de 1999.

[2] J. Neider, T. Davis, y M. Woo. "OpenGL Programming Guide". Addison - Wesley. 1993.

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

[4] K. Akeley, P. Haeberli, y D. Burns. tomesh.c: Programa en C en el Developer's Toolbox CD, 1990.

[5] F. Evans, S. S. Skiena, y A. Varshney. "Optimizing Triangle Strips for Fast Rendering". IEEE Visualization. Octubre de 1996. Pag: 319 - 326.

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

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

[8] B. Speckmann y J. Snoeyink. "Easy triangle strips for TIN terrain models". 9th Canada Conference on Computational Geometry. pag: 239 - 244. 1997.

[9] H. Hoppe. "Optimization of mesh locality for transparent vertex caching". Computer Graphics (SIGGRAPH 1999 Proceedings), pag: 269 - 276.

[10] G. Brassard, y P. Bratley. "Fundamentos de algoritmia". Prentice Hall. 1997.

[11] M. Garland, y P. Heckbert. "Surface Simplification Using Quadric Error Metrics". SIGGRAPH. Agosto de 1997. Pag: 209 - 216.

[12] D. Luebke y C. Erikson. "View-Dependent Simplification of Arbitrary Polygonal Environments". Proc. Siggraph 97. ACM Press, New York, 1997, pag:199-208.

[13] H. Hoppe. "View-dependent refinement of progressive meshes". In Proceedings of SIGGRAPH '97, Computer Graphics Proceedings, Annual Conference Series, pag: 189-197. ACM SIGGRAPH, ACM Press, Agosto 1997.

[14] Jihad El-Sana et al. "Skip Strips: Maintaining Triangle Strips for View-dependent Rendering" Proceedings of the conference on Visualization '99. 1999, pag: 131 - 138.

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

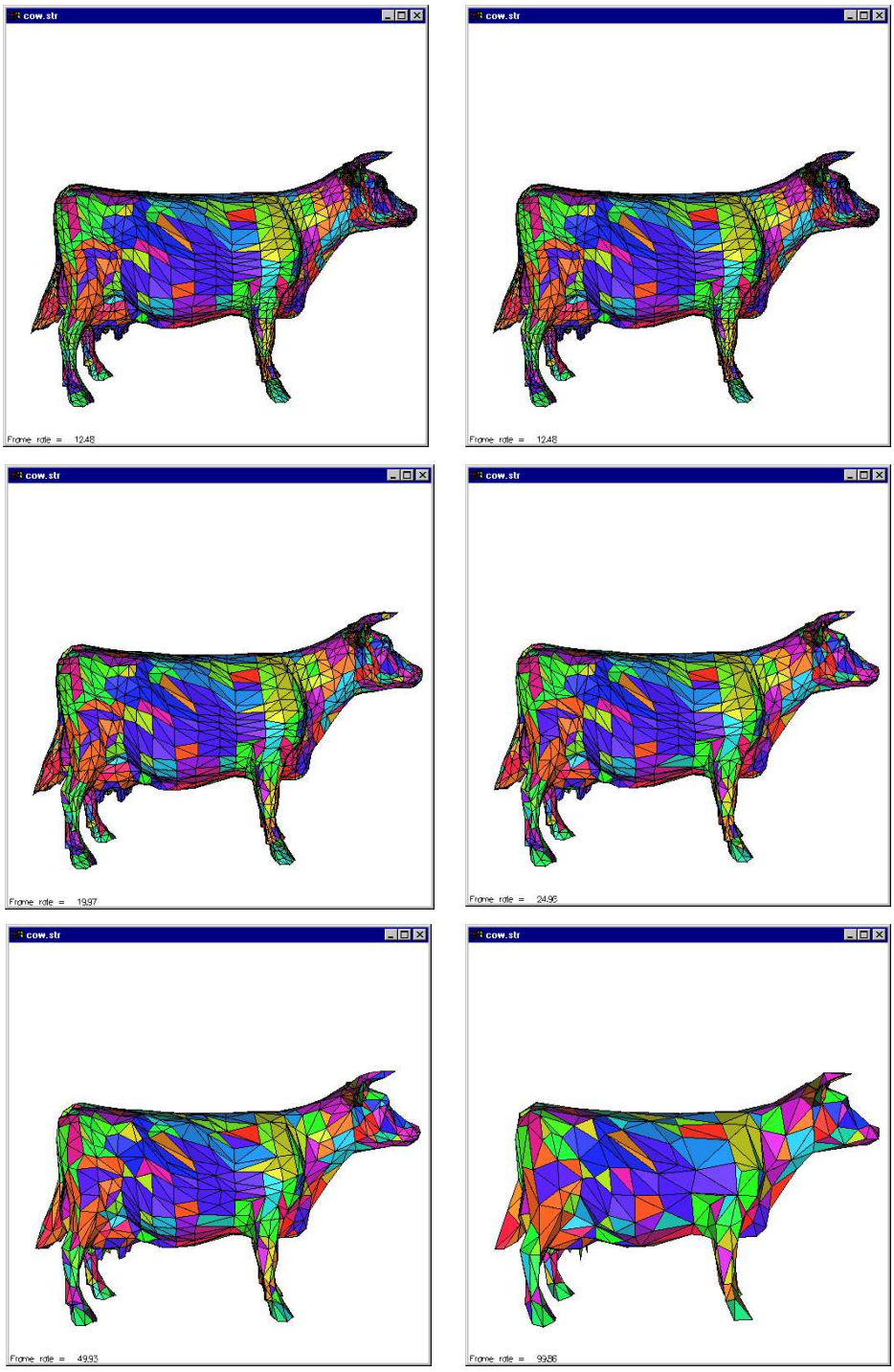

Figura 6: Tiras encontradas en un modelo poligonal y las resultantes tras colapsar 500, 1000, 1500, 2000 y 2500 aristas.

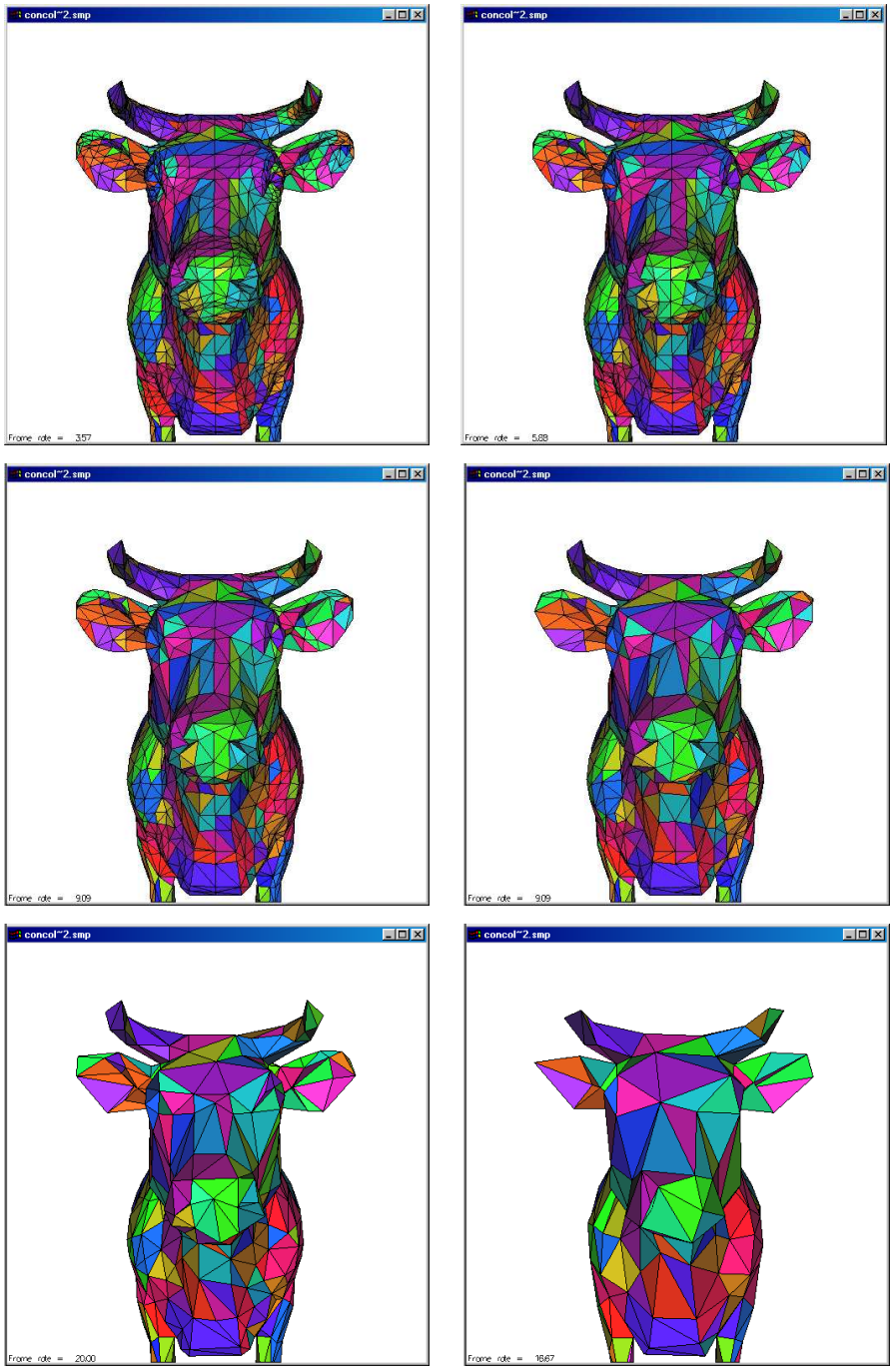

Figura 7: Detalle de la cabeza. Modelo original y resultante tras contraer 500, 1000, 1500, 2000 y 2500 aristas.