

Multirresolución adaptativa de mallas triangulares basado en criterios de textura

Alexánder Ceballos*, Jorge Hernández**, Flavio Prieto ***

Resumen

Los modelos 3D están generalmente compuestos por miles de polígonos. En ocasiones estas representaciones pueden obtenerse con la misma calidad visual pero con un menor número de polígonos. En este artículo se propone un método para reducir el tamaño de imágenes 3D texturadas basadas en mallas triangulares, conservando la calidad visual del modelo. Se introduce un criterio de textura que controla el proceso de decimación triangular. Para eliminar puntos sin necesidad de realizar una nueva triangulación, sobre la nube de puntos se usa un algoritmo poligonal de decimación. Para definir cuáles puntos deben ser removidos, se usa un filtro de Sobel 2D sobre la textura correspondiente. Se muestra que se puede usar el algoritmo para reducir los tiempos de carga, de renderización, de transferencia y de almacenamiento de una imagen 3D texturada.

Palabras claves: Multirresolución, textura, niveles de detalle, colapso de triángulos.

Abstract

Usually, 3D models are composed by thousands of polygons. Some times, those representations can be obtained with the same visual quality but with a smaller number of polygons. In this paper, we present a method that reduces the size of 3D textured images based on triangular meshes, keeping the visual quality of the model. We introduced a texture criterion that controls the triangle decimation process. We used a polygonal algorithm of decimation that permits the structured point elimination without carrying out a new triangulation on the point cloud. In order to

Fecha de recepción: 4 de julio de 2007
Fecha de aceptación: 17 de diciembre de 2007

* Ingeniero Electrónico, Universidad Nacional de Colombia, Sede Manizales y miembro del Grupo de investigación en Percepción y Control Inteligente. aceballos@unal.edu.co

** Magister en Ingeniería - Automatización Industrial, Universidad Nacional de Colombia, Sede Manizales. jorge.hernandez@cmm.ensmp.fr

*** Profesor del Departamento de Ingeniería Eléctrica, Electrónica y Computación, Universidad Nacional de Colombia, Sede Manizales. faprieto@unal.edu.co

Correspondencia: Universidad Nacional de Colombia, Cra. 27 N° 64-60, Ing. Electrónica, Manizales (Colombia).

determine which points must to be removed, we used a 2D Sobel filter on the texture. We show that the algorithm can be used for reducing the load, rendering, transfer and storage times of 3D textured images.

Key words: Multiresolution, texture, level of detail, triangle collapse.

I. INTRODUCCIÓN

La visión por computador y la computación gráfica son dos campos que se han venido combinando gradualmente, lo cual ha dado como resultado la realidad virtual [1]. Más y más aplicaciones utilizan métodos de la visión por computador para construir modelos computacionales con datos de objetos reales. Una fase crucial en el mejoramiento de la calidad visual de modelos 3D es la asignación de una textura sobre los modelos computacionales de los objetos; incluso, puede compensar en parte las imposiciones geométricas en las reconstrucciones de los modelos 3D como concavidades ocultas que no pueden digitalizarse [2].

Uno de los mayores desafíos en visión artificial es conservar la calidad de las escenas reduciendo el costo computacional [3]. En imágenes 3D puede lograrse, por ejemplo, al reducir el número de polígonos para aquellos objetos ubicados lejos del punto de vista del observador. También, para objetos adquiridos con diferente *hardware*, el sobre-muestreo puede hacer que exista información innecesaria, lo cual se puede simplificar sin que el ojo humano distinga la diferencia. Estas simplificaciones reciben el nombre de niveles de detalle [4].

La extracción de características de una superficie 3D se realiza principalmente para la segmentación y reconocimiento de objetos [5,6]. Entre las características principales se destacan la curvatura media y gaussiana, aunque también en los últimos años ha tenido auge el uso de *wavelets*, tanto para la simplificación de la superficie como para la caracterización [7-9]. Para la generación de los niveles de detalle, los algoritmos de multirresolución pueden ser de refinamiento, que comienzan con una aproximación tosca y van agregando detalles [10]; o de decimación, que empiezan con una superficie fina y van removiendo elementos [4]. También pueden ser clasificados como no poligonales (de modelos volumétricos [11], de modelos basados en imágenes [9], de superficies paramétricas, *wavelets* [12]) o poligonales (que son los más usados).

En la simplificación poligonal la idea es reducir el número de triángulos, vértices, bordes, huecos, túneles o cavidades, normalmente agrupando puntos y haciendo necesario, casi siempre, una nueva triangulación [13,14,3]. Existe una aproximación muy popular en la cual no es necesaria una nueva triangulación, debido a que trabaja directamente sobre la malla poligonal, llamada colapso de triángulos. La idea es remover puntos pertenecientes al mismo triángulo y conservar la triangulación de los triángulos vecinos.

En este trabajo se pretende simplificar la malla teniendo en cuenta información de textura, como característica para realizar simplificación en las regiones que consideramos brindan menos información.

El documento está organizado como sigue: La Sección II introduce el método usado para la simplificación de la malla y el criterio de textura usado para la conservación de información; la Sección III describe detalladamente los algoritmos usados en cada etapa, finalmente los resultados y su análisis son mostrados en la Sección IV.

II. MÉTODO ADAPTATIVO DE MULTIRRESOLUCIÓN

El objetivo es reducir el número de puntos y triángulos de un modelo 3D texturado, tratando de conservar la calidad visual, usando un criterio de textura. El método implementado para la simplificación de la malla se presenta en la Sección II-A, mientras que el criterio de textura en la Sección II-B.

A. Método de multirresolución

El método implementado es el colapso de triángulos debido a las ventajas que éste ofrece [15, 16]: la posibilidad de optimización escogiendo qué vértices son removidos dadas las características deseadas; no es necesaria una nueva triangulación, la malla varía sobre sí misma al removerse los puntos con los bordes y triángulos ligados a ellos; simplicidad, rapidez y efectividad, no se requiere gran cantidad de cálculos para escoger la nueva posición del punto, se trabaja directamente sobre los datos originales sin necesidad de transformaciones ni funciones complejas.

- 1) *Colapso de Triángulos*: Los tres vértices de un triángulo son unidos en uno (Figura 1). Al desaparecer el triángulo, lo hacen también los tres triángulos adyacentes a sus lados, es decir que el número total de puntos decrece en dos, mientras el de triángulos lo hace en cuatro (Figura 1).

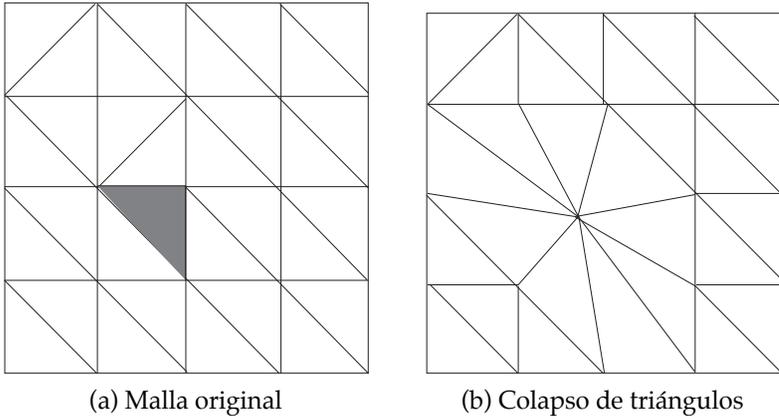


Figura 1. Colapso de triángulos

Para colapsar triángulos se debe tener en cuenta cuáles puntos pueden ser unidos. Considérese los siguientes casos:

- Colapsando todos los vecinos en un solo ciclo, y teniendo en cuenta que cada punto pertenece a más de un triángulo, no se simplifica la malla, lo único que se consigue es cambiar la posición espacial de los puntos; sin embargo, la complejidad de la malla es similar.
- Colapsando iterativamente todos los triángulos, se obtiene un punto al final, debido a que cada punto nuevo pertenece a nuevos triángulos; si no se mantiene información de cuáles ya han sido removidos se obtiene un resultado parecido al de la Figura 2.

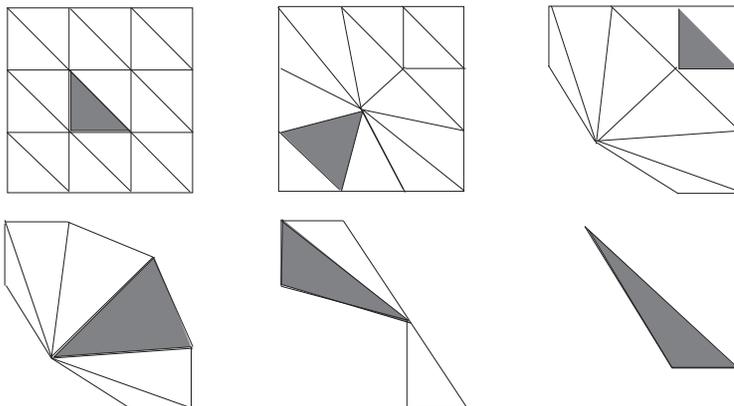


Figura 2. Colapso iterativo de triángulos

B. Estimación del criterio de textura

Cuando un modelo 3D se encuentra texturado, una función de parametrización asocia la información de textura con la nube de puntos del modelo. Esta función de parametrización relaciona la información de cámara (Ecuación 4) y la transformación de cuerpo rígido (Ecuación 5) de cada una de las vistas.

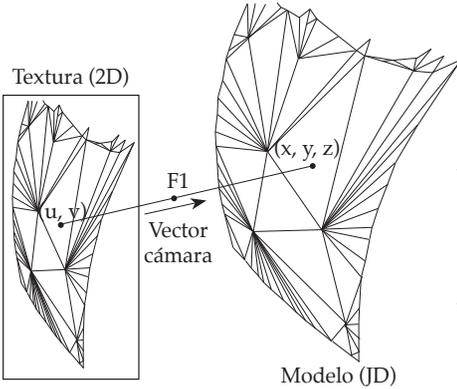
$$f(\hat{x}, \hat{y}, \hat{z}) = \begin{cases} u = \frac{p_{k00}\hat{x} + p_{k01}\hat{y} + p_{k02}\hat{z} + p_{k03}}{p_{k20}\hat{x} + p_{k21}\hat{y} + p_{k22}\hat{z} + p_{k23}} \\ v = \frac{p_{k10}\hat{x} + p_{k11}\hat{y} + p_{k12}\hat{z} + p_{k13}}{p_{k20}\hat{x} + p_{k21}\hat{y} + p_{k22}\hat{z} + p_{k23}} \end{cases} \quad (4)$$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = (T_i)^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5)$$

A partir de la textura asignada se puede establecer cuáles triángulos del modelo no pueden ser eliminados en el proceso de decimación. Las regiones en la imagen con alta variación son relacionadas con los triángulos en el modelo. Sin embargo, existen infinitos puntos en el espacio que se proyectan en un mismo píxel en la textura; por esta razón, se estableció un método con el cual se asignan los puntos sobre el modelo 3D.

Con el conocimiento de la función de parametrización de la textura se determina un punto en el espacio (P_i) y el vector de cámara con los cuales se determina la línea paramétrica de los puntos que se proyectan en el mismo píxel. Con la ecuación de la línea paramétrica, se intercepta cada uno de los triángulos visibles del modelo. Finalmente, el punto perteneciente a un triángulo es el punto 3D proyectado en la textura. La Figura 3 ilustra el proceso donde se determina un punto sobre la imagen de rango a partir de un píxel de la textura.

De esta forma, para todos los píxeles correspondientes a regiones de alta variación se encuentra el triángulo correspondiente en el modelo 3D.



Línea paramétrica
 $L(u) = P_i + u \cdot (V_{camara})$

Punto de corte de una línea paramétrica y un plano

$$P = P_i + \frac{(N_{plano} \cdot (P_{plano} - P_i))}{(N_{plano} \cdot V_{camara})} \cdot V_{camara}$$

Figura 3. Asignación de un punto sobre la imagen de rango a partir de un píxel de la textura

III. IMPLEMENTACIÓN

A partir de un modelo 3D, con la textura asociada a cada uno de los triángulos que lo componen, se ejecuta el Algoritmo 1. Para la extracción del criterio de textura para cada punto de la nube se emplea el método descrito en la Sección II-B. De esta forma se establece cuáles triángulos no pueden ser removidos. En la etapa de decimación se utiliza el algoritmo de colapso de triángulos descrito en la Sección II-A. Finalmente, para la medida del error se propone el uso de la distancia de cada uno de los puntos del modelo original al triángulo más cercano de modelo decimado, propuesto por Heok en [4].

Algoritmo 1. MULTIRRESOLUCIÓN

Identificación de los puntos que conforman el borde: Bordes

Estimación del criterio de textura: Textura.

Simplificación de la malla del modelo: Decimación.

A continuación se describe cada uno de los procesos realizados en la implementación.

A. Extracción de bordes

La identificación de los puntos que conforman los bordes es necesaria para evitar que sean removidos y se deforme la malla. Una vecindad “sombriila”

es una región compuesta por los triángulos que rodean a un punto. Para que un punto sea considerado borde, su vecindad “sombriilla” debe estar incompleta. En una vecindad “sombriilla” completa cada punto debe estar compartido por mínimo dos triángulos. En el Algoritmo 2, para la obtención de los puntos vecinos al actual, se incluyen en un arreglo los otros dos vértices de los triángulos que lo comparten, de forma tal que aquel punto que no se repita pertenezca sólo a un triángulo de la vecindad “sombriilla”, lo que significa que es incompleta y, por lo tanto, el punto actual es un borde. En la Figura 4, los puntos 1 y 3 sólo pertenecen a un triángulo en la vecindad “sombriilla” por lo que la vecindad está incompleta y el punto actual es un borde.

Algoritmo 2. BORDES

Obtención de los puntos vecinos a cada punto.

Si el punto no tiene vecinos

Entonces es un borde

Fin Si

Si cada punto vecino no se repite en la vecindad

Entonces es un borde

Fin Si

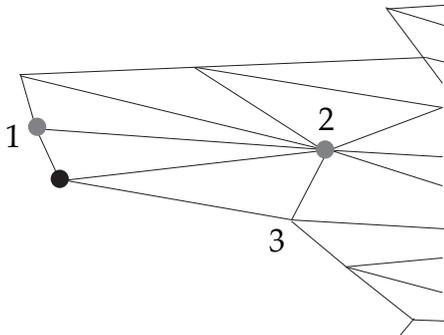


Figura 4. Punto borde

B. Extracción del criterio de textura

En la implementación y al igual que para la extracción de bordes, el análisis del criterio de textura se lleva a cabo para cada punto de la malla triangular.

En el caso de una imagen de rango, la transformación de cuerpo rígido es omitida, y para determinar las regiones de alta variación sobre la textura asignada se utilizó un filtro de Sobel normalizado sobre la textura correspondiente. De esta forma, se le asigna a cada triángulo el valor de cada píxel filtrado. Así, los puntos que presenten un valor alto en el filtro de Sobel son puntos que presentan, a su vez, un valor alto en el criterio de textura.

C. Decimación

La decimación consiste en la simplificación de la malla removiendo primitivas. Se remueven triángulos y puntos usando el colapso de triángulos (Algoritmo 3). Para conseguir el objetivo propuesto, no se simplifican aquellos puntos que son bordes, ni cuyo criterio de textura es alto. Debido a que el colapso debe ser hecho sobre puntos vecinos que pertenecen a un mismo triángulo, el análisis se hace para cada triángulo de la malla.

Para evitar simplificar la malla a un solo punto (Figura 2) se usan variables booleanas que definen cuáles ya han sido removidos, y lo más importante, cuáles no pueden ser borrados.

Al realizar un colapso de triángulos se eliminan los tres vértices que lo componen y los triángulos que poseen dos de ellos; se conservan los triángulos que sólo poseían uno de los puntos colapsados y no deben ser removidos en la siguiente iteración como se puede observar en la Figura 1.

Algoritmo 3. DECIMACIÓN

Para todos los triángulos **Hacer**

Si ni ningún vértice es borde **y** tienen criterio bajo **y** el triángulo puede ser removido **y** no ha sido borrado

Entonces $d = (a + b + c) / 3$

Los vértices son reemplazados por d

El triángulo **y** todos los que compartían dos puntos con él son borrados

Los triángulos que compartían un vértice con el triángulo no pueden ser removidos

Fin Si

Fin Para

Corrección de algunas deformaciones que aparecen en la malla del modelo decimado: corregir

Debido a que se modifica la malla sobre sí misma, se presentan algunos casos particulares en los que la malla se deforma de manera indeseable. Algunas veces, dos triángulos vecinos después de todo el proceso de decimación terminan compartiendo los tres vértices, pareciendo una espiga que sobresale de la malla y, por lo tanto, ambos deben ser removidos.

También existe la posibilidad de que se forme una especie de pirámide en la cual un triángulo parece estar en orientación contraria a la de los triángulos que le rodean (Figura 5); esto puede ser solucionado al eliminar los tres triángulos y reemplazarlos por uno solo.

Sin embargo, aún se hace necesario cambiar la orientación de algunos triángulos, cuya orientación es contraria debido a la manipulación de sus vértices; esto se logra al cambiar el orden de los puntos que componen el triángulo (Algoritmo 4).

Algoritmo 4. CORREGIR

Remoción de espigas

Para todos los triángulos **Hacer**

Si algún triángulo vecino posee los mismos puntos

Entonces ambos son borrados

Fin Si

Fin Para

Remoción pirámides

Para todos los puntos **Hacer**

Si pertenece a tres triángulos y el ángulo entre uno de los triángulos y los otros dos es mayor a 90°

Entonces: borrar los tres y conformar uno nuevo

Fin Si

Fin Para

Cambiar la orientación de los triángulos que están al revés

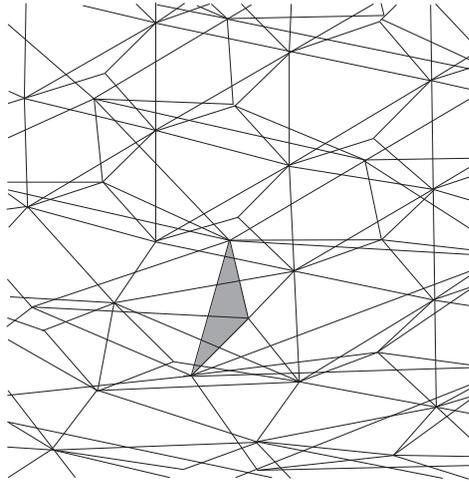


Figura 5. Corrección de malla

IV. RESULTADOS

Los experimentos se realizan sobre imágenes de rango con textura de un rostro femenino (Figura 6 y Figura 9), de un rostro masculino (Figura 7 y Figura 10), y de una escultura precolombina (Figura 8 y Figura 11). Los resultados se compararon con los obtenidos tras aplicar el mismo principio, pero usando un criterio de curvatura para definir que regiones son removidas, como se propuso en [16]. Además, para el cálculo del error propuesto en esta sección, se usaron los tres modelos normalizados estadísticamente.

A. Decimación con criterio de textura

En la Figura 6 se observa la decimación con criterio de textura realizada sobre una imagen de rango de un rostro femenino de 222.463 triángulos, sigue siendo aceptable en la quinta iteración con 32.351 triángulos, como se puede ver tanto en el modelo completo como en la malla triangular. También en la Figura 7 se observa la decimación con criterio de textura realizada sobre una imagen de rango de un rostro masculino de 209.222 triángulos, la cual sigue siendo aceptable en la tercera iteración con 30.083 triángulos. Finalmente, se muestra la decimación con criterio de textura de una imagen de rango de una escultura precolombina de 138.807 triángulos (Figura 8), el modelo sigue siendo aceptable en la cuarta iteración con 18.635 triángulos, como se observa tanto en el modelo completo como en la malla triangular.

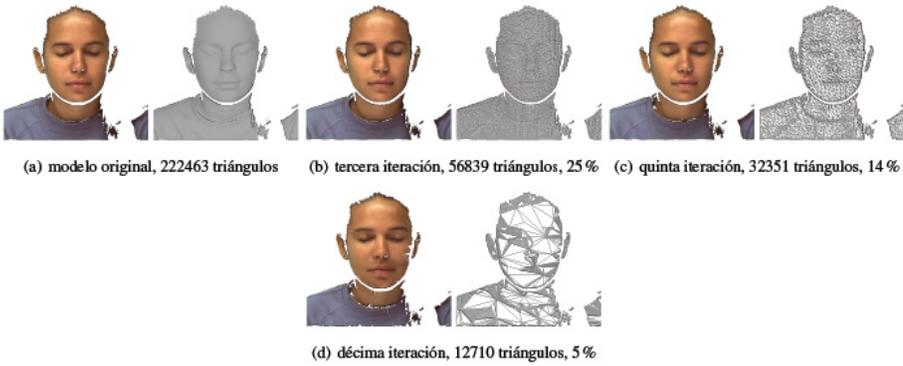


Figura 6. Decimación con criterio de textura realizada sobre una imagen de rango con textura de un rostro femenino de 222.463 triángulos

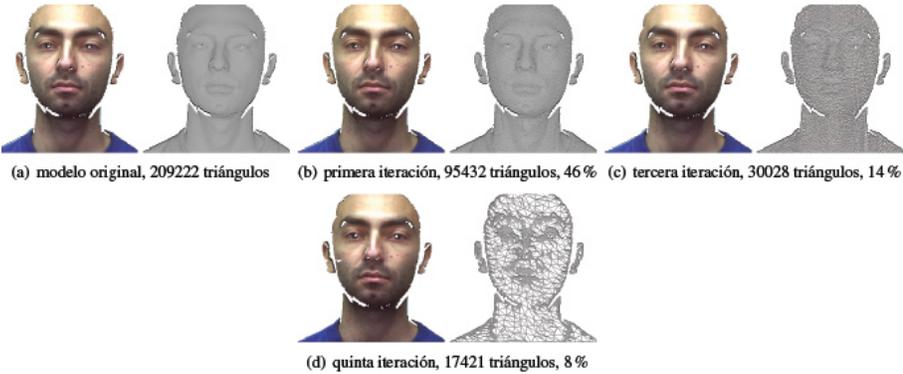


Figura 7. Decimación con criterio de textura realizada sobre una imagen de rango con textura de un rostro masculino de 209.222 triángulos

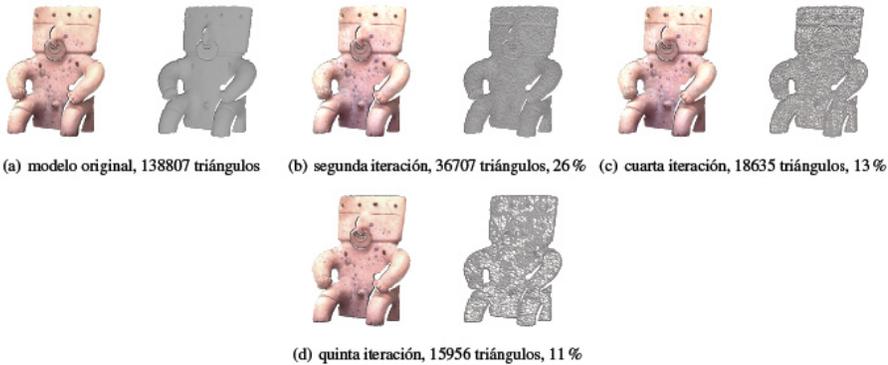


Figura 8. Decimación con criterio de textura realizada sobre una imagen de rango con textura de una escultura precolombina de 138.807 triángulos

B. Decimación con criterio de curvatura

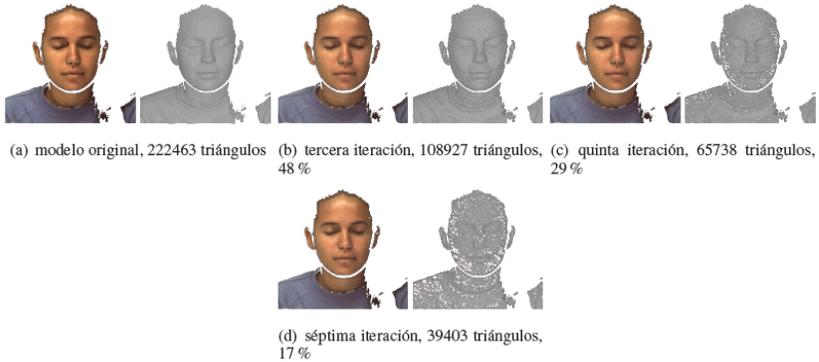


Figura 9. Decimación con criterio de curvatura realizada sobre una imagen de rango con textura de un rostro femenino de 222.463 triángulos

En la Figura 9 se observa la decimación con criterio de curvatura realizada sobre una imagen de rango con textura de un rostro de 222.463 triángulos; el modelo sigue siendo aceptable en la séptima iteración con 39.403 triángulos, como se puede ver tanto en el modelo completo como en la malla triangular. También en la Figura 10 se observa la decimación con criterio de curvatura realizada sobre una imagen de rango con textura de un rostro masculino de 209.222 triángulos, de nuevo, el modelo sigue siendo aceptable en la séptima iteración con 38341 triángulos. Finalmente, se muestra la decimación con criterio de curvatura de una imagen de rango con textura de una escultura precolombina de 138.807 triángulos (Figura 11), nuevamente, el modelo sigue siendo aceptable en la séptima iteración con 23.714 triángulos, como se puede ver tanto en el modelo completo como en la malla triangular.

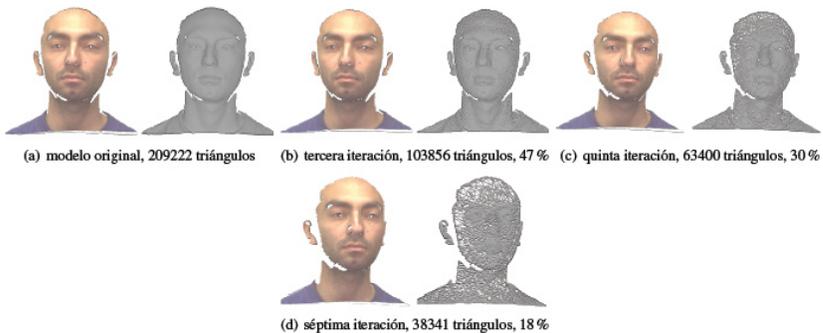


Figura 10. Decimación con criterio de curvatura realizada sobre una imagen de rango con textura de un rostro masculino de 209.222 triángulos

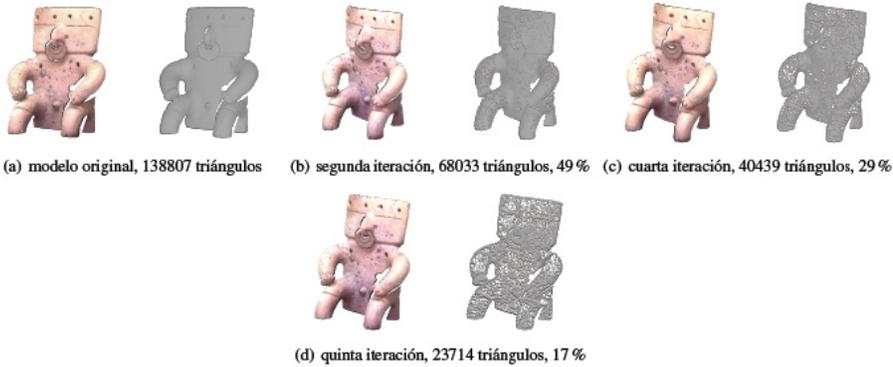


Figura 11. Decimación con criterio de curvatura realizada sobre una imagen de rango con textura de una escultura precolombina de 138.807 triángulos

Para la medida del error, se propone usar la distancia de los puntos del modelo original a los triángulos del decimado, para lo cual se normalizan estadísticamente los modelos. Se realizan 30 iteraciones para el método mediante criterio de textura y 30 para el método mediante criterio de curvatura para los tres modelos, y se obtuvieron los resultados que se muestran en las Figuras 13 y 14.

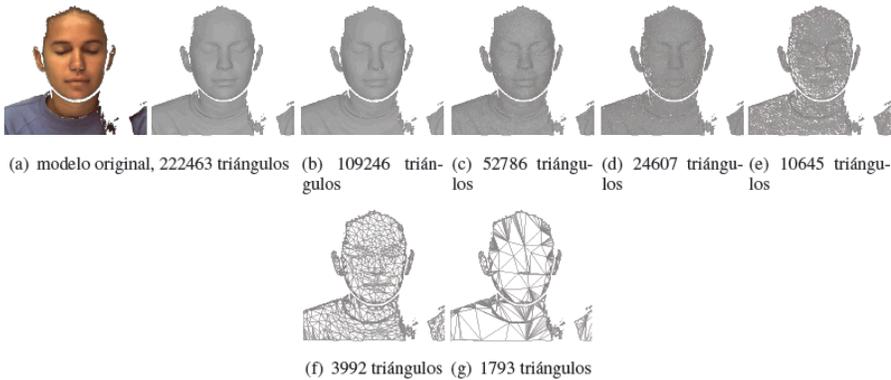


Figura 12. Decimación realizada con *Polygon Editor* sobre una imagen de rango de un rostro femenino de 222.463 triángulos

Para los diferentes modelos el error tiene un comportamiento similar respecto a la relación de triángulos eliminados, tanto para el caso en el que se usa el criterio de textura como el de curvatura, y se pudieron aproximar a las curvas que se muestran en las Figuras 13 y 14. Como se ve, la medida del error crece drásticamente después de haber sido eliminado el 80% de triángulos.

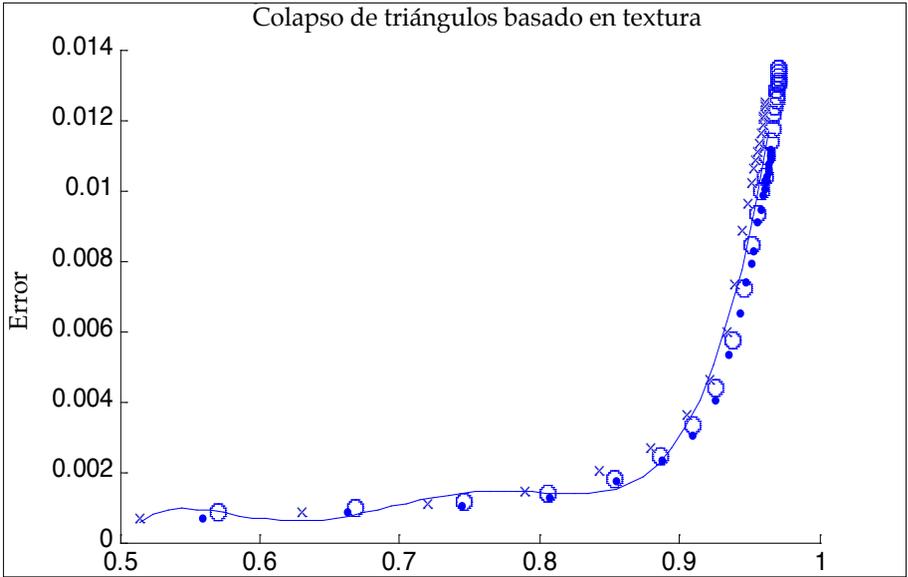


Figura 13. Medida del error para tres modelos mediante el método de decimación basado en textura

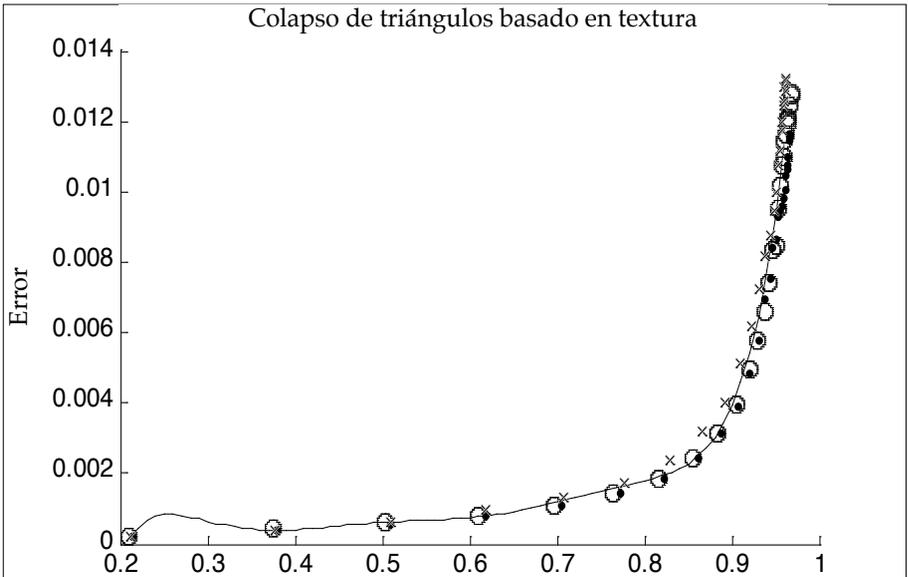


Figura 14. Medida del error para tres modelos mediante el método de decimación basado en curvatura

Se compararon con los resultados obtenidos al decimar el modelo con el *Polygon Editor*, un *software* especializado, como se observa en la Figura 15.

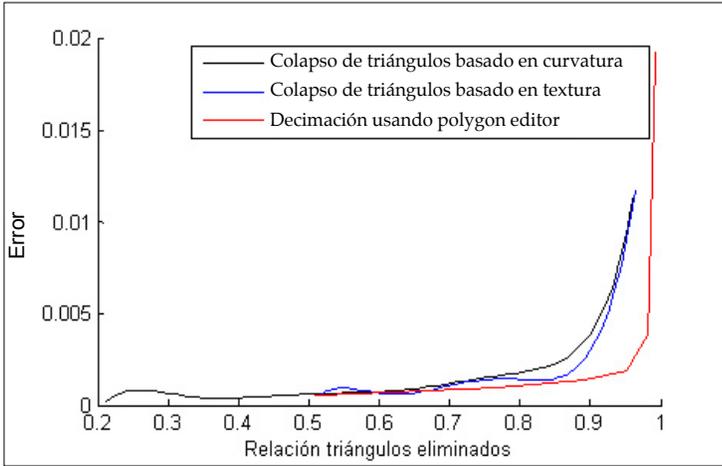


Figura 15. Comparación de la medida del error para la decimación con criterio de textura, de curvatura y usando el *Polygon Editor*.

La menor medida de error se obtiene con el *Polygon Editor*, pero las aproximaciones propuestas son buenas antes de que se elimine el 80% de los triángulos.

El algoritmo converge rápidamente, como se observa en la Figura 16; en la décima iteración ya se ha removido cerca del 90% de los triángulos originales, y en las iteraciones siguientes el número se reduce lentamente.

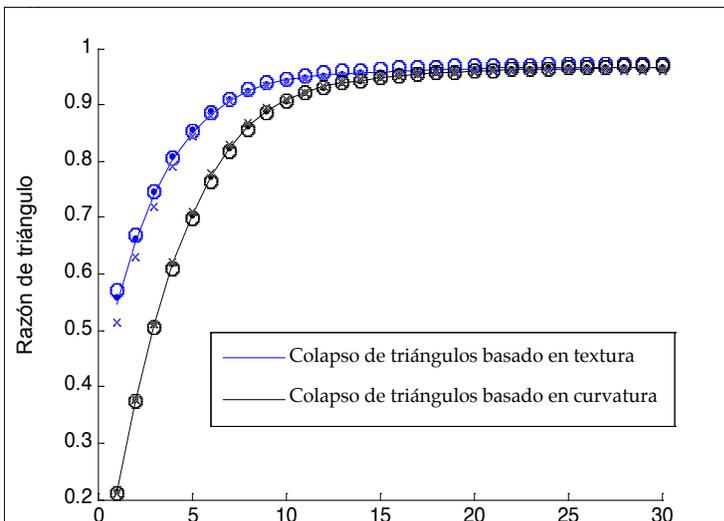


Figura 16. Relación de triángulos eliminados vs número de iteraciones

V. CONCLUSIONES

Se puede reducir el número de puntos y triángulos de un modelo 3D, usando criterios de textura para una imagen de rango, y aplicando un algoritmo de decimación en aquellas zonas que representan poca información para reducir el costo computacional, de almacenamiento y de renderización.

Una de las mayores limitaciones de los algoritmos propuestos es que se presentan algunas deformaciones no deseadas sobre la malla, debido a que se trabaja directamente sobre ella sin necesidad de realizar una nueva triangulación; aunque esto hace que el algoritmo sea rápido, podrían obtenerse mejores resultados. El problema radica en el proceso de triangulación; se propone implementar otro algoritmo de decimación en el cual sea necesario realizar una nueva triangulación para comparar resultados y tiempos de cómputo

Aunque un algoritmo de decimación en el cual no es necesario un postproceso de triangulación presenta falencias en cuanto a la forma de la malla, tiene ventajas en complejidad computacional y es adecuado para aplicaciones de renderización como plataformas web y juegos.

Agradecimientos

Los autores agradecen a la Universidad Nacional de Colombia, por su apoyo en el marco del proyecto titulado “Modelado de Superficies de Forma Libre Empleando Técnicas de Visión Artificial”.

REFERENCIAS

- [1] LAURENDEAU, D. BERTRAND, N. y HOUDE, R. The Mapping of Texture on vr Polygonal Models. En: Second International Conference on 3-d imaging and modelling (1999). Proceedings. Second International Conference on Volume 3-D Digital Imaging and Modeling. Ottawa, Canada , 1999, pp 332-339.
- [2] HERNÁNDEZ, Jorge. Mapeo de texturas a objetos 3D basado en la geometría de la escena. Tesis (Magíster en Ingeniería). Manizales, Colombia. Universidad Nacional de Colombia Sede Manizales, 2006.
- [3] CHACON, D. Estudio y análisis de la teoría de la multirresolución en el modelado de sólidos. Tesis (Magíster en Ciencias con Especialidad en Ingeniería en Sistemas Computacionales). Puebla, México. Universidad de Las Américas-Puebla, 2000.
- [4] TAN, Heok Kim y DAMAN, D. A review on level of detail. En: International Conference On Computer Graphics, Imaging And Visualization (2004). CGIV

2004. Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. pp. 5.
- [5] PAPAIOANNOU, Georgios, KARABASSI, Evaggelia-Aggeliki y THEOHARIS, Theoharis. Segmentation and surface characterization of arbitrary 3D meshes for object reconstruction and recognition. En: 15TH International Conference on Pattern Recognition (2000). ICPR'00. Proceedings. 15th International Conference on Pattern Recognition. University of Athens, GREECE, 2000, Volume 1 p. 1734.
- [6] BERNAL, Claudio y DOMÍNGUEZ, Manuel. Reconocimiento y modelización automatizada de instalaciones industriales normalizadas mediante procesos de ingeniería inversa. En: Congreso Internacional Conjunto XV Adm – XVIII Ingeggraf (2005). Memorias. Congreso Internacional Conjunto XV ADM – XVIII INGEGRAF. Universidad de Sevilla, Sevilla, 2005.
- [7] SHAFFER, Eric y GARLAND, Michael. A multiresolution representation for massive meshes. *IEEE Transactions on Visualization and Computer Graphics*, 11(2), 2005.
- [8] CASTRO, S. CASTRO, L. DE GIUSTI, A. Multiresolution volume representation: the wavelet approach. En: International Conference of the Chilean Computer Science Society (XIX, 1999, Chile). Proceedings. XIX International Conference of the Chilean Computer Science Society), Talca, Chile, SCCC, 1999.
- [9] GU, X. GORTLER, S y HOPPE, H. Geometry Image. En: Annual Conference Series in Computer Graphics (2002). SIGGRAPH'02. Proceedings. Annual conference series in computer graphics. New York, USA, SIGGRAPH, 2002, pp 355-361.
- [10] MERY, Domingo. Visión por computador. Santiago de Chile. Universidad Católica de Chile. 2004, 138 p.
- [11] BAYONA, S. Berilo y GARCÍA, M. Lorenzo. Estudio y análisis de la multirresolución en el modelado de sólidos. En: International Conference on Pattern Recognition (12, 2002, Quebec). ICPR 2002. Proceedings. International Conference on Pattern Recognition,. Quebec, Canada, ICPR, 2002.
- [12] PASTOR, Luis, RODRÍGUEZ, Ángel. Surface approximation of 3D objects from irregularly sampled clouds of 3D points using spherical wavelets. En: International Conference on Pattern Recognition (10, 1999, Washington). ICIAP 1999. Proceedings. 10th International Conference on Image Analysis and Processing, Washington, USA, ICIAP, 1999.
- [13] ROSSIGNAC, Jarek y BORREL, Paul. Multi-resolution 3D approximations for rendering complex scenes. Springer-Verlag. Modeling in Computer Graphics: 455-465, 1993
- [14] KALVIN, Alan and TAYLOR, R. H. Surfaces: Polygonal mesh simplification with bounded error. *IEEE Computer Graphics & Applications*. 16(3):64-77. 1996.
- [15] GIENG, T.S., HAMANN, B. JOY, K.I., SCHLUSSMANN, G.L. y TROTTS, I.J. Smooth Hierarchical Surface Traingulations, En: Yagel, R. and Hagen, H. (Eds.). IEEE Visualization '97. Phoenix, Arizona: IEEE. 379-386.
- [16] CEBALLOS, A., HÉNANDEZ, J. y PRIETO, F. Multirresolución adaptativa de mallas triangulares basado en criterios de curvatura. En: Congreso Colombiano de Computación (2, 2007, Bogotá). 2CCC. Memorias. Segundo Congreso Colombiano de Computación. Bogotá, Colombia. 2CCC, 2007.