

HERRAMIENTAS DE MODELADO DE DATOS LÁSER: CREACIÓN, RELLENO DIEZMADO, Y COMPARACIÓN DE MALLAS.

SANTIAGO FABADO SALVADOR

Universitat Politècnica de València. Grupo de Investigación de Fotogrametría y Láser Escáner (*GIFLE*)
sanfasal@topo.upv.es

MIRIAM CABRELLES LÓPEZ

Universitat Politècnica de València, *GIFLE*
micablo@upvnet.net

ANA ELENA SEGUÍ GIL

Universitat Politècnica de València, *GIFLE*
ansegil@upvnet.net

JOSÉ LUIS LERMA GARCÍA

Universitat Politècnica de València, *GIFLE*
jllerma@cgf.upv.es

Resumen: Desde hace varios lustros se dispone de la tecnología láser escáner que permite digitalizar en 3D cualquier superficie de una manera revolucionaria, a partir de la medición masiva de puntos. Ante tal cantidad de información en bruto, es necesaria la creación de sistemas de gestión de dichos datos para poder manejarlos y extraer información válida y relevante.

Este artículo presenta una aplicación informática propia de modelado o reconstrucción de datos láser. Como datos de partida, se parte de una nube de puntos capturada mediante láser escáner terrestre, el *ninot* de una falla valenciana. A continuación, se procede a la generación de la triangulación 3D siguiendo procedimientos automáticos, tratando de asegurar una correcta reconstrucción de la compleja realidad física. Se presentarán diferentes algoritmos como son la triangulación progresiva (basado en la triangulación Delaunay 3D), el relleno de agujeros y el diezmado de la triangulación. Para finalizar el estudio, se realizará una comparación entre el modelo original y el diezmado.

Palabras claves: *3D Delaunay, diezmado, comparación de mallas, modelado, láser escáner, reconstrucción, triangulación.*

Abstract: Since a few decades ago, laser scanner technology is available and allows users the digitalization of whatever surface in a revolutionary way, measuring massively the three-dimensional points that define such surface. Given such a huge amount of raw data, it is required to create management systems in order to extract the valid and relevant information.

This paper presents a computer application for modeling and reconstruction laser scanner data. It starts from a point cloud captured by the terrestrial laser scanner, a *ninot* of a Valencian falla. Afterwards the generation of the 3D mesh starts following automation, and attempting to ensure the correct reconstruction of the complex physical reality. Several

algorithms will be presented such as progressive triangulation (based on the 3D Delaunay triangulation), hole filling, and mesh decimation. Finally, a comparison of the different obtained models will be carried out to verify the discrepancies between the triangulations.

Key words: *3D Delaunay, decimation, mesh comparison, modelling, laser scanner, reconstruction, triangulation.*

1. INTRODUCCIÓN

En la actualidad, los modelos 3D son demandados en ámbitos diversos tales como la arquitectura, la arqueología, la ingeniería industrial, la animación virtual, y en aquellos videojuegos que incorporan entornos virtuales. Técnicas y tecnologías como son la fotogrametría a partir de imágenes y el escáner láser permiten la medición masiva de puntos sobre una superficie, clave para definir la forma y el tamaño del objeto medido con gran precisión, dando como resultado una nube de puntos densa. Sin embargo, en la mayoría de casos es necesario procesar esa nube de puntos para transformarla en algo que nos sea útil, y ello pasa por crear una superficie continua que enlace todos esos puntos. Este proceso es conocido como reconstrucción, mallado o modelado de nubes de puntos.

En este artículo se presenta una aplicación informática propia, que prima la calidad y el acabado del modelo, y que consta de subprocesos como son la triangulación progresiva, el relleno de agujeros y el diezmado de la malla o modelo. En primer lugar, tanto la triangulación progresiva como el relleno de agujeros están basados en la triangulación Delaunay 3D. La triangulación Delaunay [HIL96] se ha convertido en uno de los principales métodos para la reconstrucción 3D en aplicaciones informáticas, puesto que garantiza la calidad en la malla creada al maximizar los

ángulos interiores de cada triángulo. En segundo lugar, el método seguido para el diezmo de la malla sigue el concepto comúnmente denominado como colapso de triángulos [PAN01], categorizado como un algoritmo de eliminación de geometría. Dicho algoritmo es valorado por su rapidez, puesto que elimina más puntos del modelo en cada iteración que otros algoritmos. Finalmente, se ha elaborado una herramienta que permite la comparación entre dos modelos triangulados. El método elegido para dicha comparación se basa en la distancia Hausdorff [ROY02], que cuantifica la longitud lineal de la separación existente entre las superficies de los dos modelos, aportando una comparación global.

En la Sección 2 de este artículo se realiza una breve descripción de la tecnología utilizada para desarrollar la herramienta. En la Sección 3 se incluye tanto el análisis de requisitos previos como la especificación de las funcionalidades de la herramienta. En la Sección 4 se presentan los resultados obtenidos con la aplicación. Y por último en la Sección 5 se establecen conclusiones acerca de las utilidades desarrolladas y la tecnología empleada.

2. TECNOLOGÍA

2.1. Láser Escáner Terrestre

El láser escáner terrestre estático es un instrumento capaz de obtener coordenadas tridimensionales de un conjunto de puntos mediante la medición de la distancia de la estación al punto, y la medición de los ángulos del haz emitido respecto al sistema de referencia propio del instrumento. Es estático porque se mantiene en una posición fija durante la adquisición de datos. El propósito de un escáner 3D es, generalmente, el crear una nube de puntos a partir de muestras geométricas en la superficie del objeto. Estos puntos se pueden usar entonces para extrapolar la forma del objeto. También se podrán incorporar los colores en la superficie del objeto si se reúne la información de color para cada uno de los puntos.

2.2. Microsoft .NET

La plataforma .NET de Microsoft es un componente de programas que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución de Microsoft pretende ser utilizada por la mayoría de las aplicaciones creadas para su sistema operativo Windows. La aplicación se sirve del lenguaje C# [KOV07], un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, cuya sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET. Asimismo, el entorno de programación ha sido Microsoft Visual Studio 2008.

3. DISEÑO TÉCNICO

3.1. Triangulación 3D

La triangulación o reconstrucción 3D [ROD01] es el proceso mediante el cual objetos reales son reproducidos en la memoria de un ordenador manteniendo sus características físicas (dimensiones, volumen y forma). El proceso realiza la conexión del conjunto de puntos representativos del objeto en forma de elementos de superficie, que en este caso son triángulos. En el diseño del proceso se ha optado por seguir un algoritmo iterativo, el cual añade los triángulos al modelo uno a uno. El proceso comienza por crear un primer triángulo, y a partir de los tres segmentos que lo componen se crean los restantes triángulos, que en su totalidad compondrán el modelo 3D. A este algoritmo se le ha denominado de expansión. De modo que para cada iteración en la construcción de un nuevo triángulo a partir de un segmento existente, el algoritmo necesita unas pautas a seguir para la elección del triángulo óptimo, pues existen un número infinito de conexiones entre los puntos. Así pues, para cada segmento nuevo, el algoritmo construye todos los posibles triángulos con los puntos existentes alrededor (Fig. 1), y evalúa cuál es el óptimo según unas condiciones establecidas, las cuales se detallan a continuación.

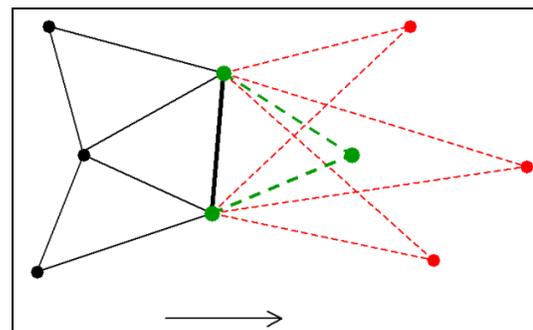


Fig. 1. Construcción de nuevos triángulos partiendo de un segmento previo.

3.1.1. Condiciones en la construcción: En este apartado se especifican y detallan las 6 condiciones que debe cumplir nuestro algoritmo para la correcta construcción de la triangulación.

- *Delaunay* [FIS07][XU10]: La triangulación de Delaunay maximiza los ángulos interiores de los triángulos del modelo. Dado un segmento formado por dos vértices, se completará el triángulo con el vértice de menor radio de la circunferencia circunscrita. En la figura 1 aparece en color verde el triángulo que cumple la condición de Delaunay.
- *Propiedad de la esfera vacía*: define que la circunferencia circunscrita de cada triángulo de la malla no debe contener ningún otro vértice. Puede ocurrir que aunque se cumpla la primera condición (Fig. 2), queden

puntos dentro de la esfera, por tanto, para la correcta construcción de la malla de triángulos, se incluirá en el algoritmo una operación que detecte si existen puntos o vértices dentro de la esfera circunscrita en el nuevo triángulo. En la figura 2 aparece en color verde el triángulo óptimo.

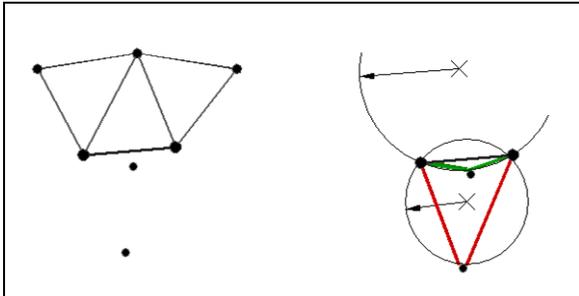


Fig. 2. Propiedad de la esfera vacía. Se muestra en color rojo el triángulo que cumple la propiedad Delaunay, y en color verde el triángulo que cumple con la propiedad de la esfera vacía.

- *Topología*: Se ha usado la topología para: lograr que las normales de los triángulos se dibujen en el mismo sentido (Fig. 3); garantizar que un segmento forme parte de dos triángulos como máximo; y por último para conocer la adyacencia entre triángulos.

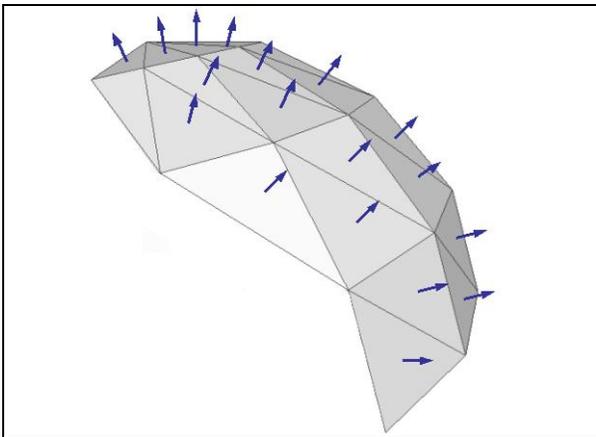


Fig. 3. Coherencia en el sentido de las normales de los triángulos.

- *Superficie continua*: un objetivo esencial es garantizar la construcción de la malla de manera continua. Esto es, que la malla de triángulos sea continua al igual que la superficie que representa. El algoritmo se basa en el concepto de expansión de malla comentado anteriormente, ver figura 1. Para ello se almacenan los segmentos que se van creando, para recuperarlos después utilizándolos como base de posteriores triángulos.

- *Ángulo máximo entre normales*: El motivo para el uso de este umbral es que en principio se desea trabajar con nubes

de puntos que formen superficies generalmente suaves, de las cuales se ha reducido el ruido, es decir, en las que no existen constantemente líneas de rotura. Por ello, el ángulo entre normales de triángulos contiguos tiende generalmente a no pasar de los 90° (Fig. 4). En cada iteración, se calcula el ángulo entre las normales del posible nuevo triángulo y del triángulo predecesor. En caso de que este ángulo supere el umbral, el algoritmo desestima ese triángulo y pasa a evaluar al siguiente.

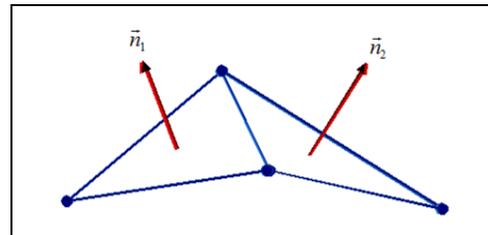


Fig. 4. Ángulo entre las normales de dos triángulos adyacentes.

- *Relleno progresivo*: Este concepto define que el umbral en el radio aumenta según se crea la malla. En primer lugar se utiliza un umbral de radio determinado por el usuario, y se comienza a triangular. Cuando se terminan de construir todos los triángulos atendiendo a ese umbral, éste se ve incrementado en un porcentaje, y se continúa construyendo. El proceso termina cuando el umbral de radio rebasa un valor determinado, o cuando todos los puntos de la nube han sido usados al menos una vez.

3.1.2. Uso del KD-Tree en el proceso: En ciencias de la computación, un Árbol kd o KD-Tree [SHA02] es una estructura de particionado de datos espaciales que organiza los puntos en un espacio euclídeo de k dimensiones. El árbol kd es un caso especial de los árboles BSP (Partición Binaria del Espacio), el cual solo emplea planos perpendiculares al sistema de coordenadas. El resultado de una estructuración de los datos en forma KD-Tree es la división del espacio 3D total que ocupa la nube de puntos en pequeños volúmenes llamados celdas. Cada celda contiene un número máximo de n puntos. Este tipo de estructuración de los datos se utiliza para ganar rapidez en el acceso a la información geométrica de la nube de puntos. Además, donde verdaderamente se gana rapidez es en el cálculo del proceso de triangulación, ya que al tener los puntos organizados en celdas se puede reducir el espacio de búsqueda. Conociendo la celda donde se encuentra un determinado punto, podemos conocer el resto de puntos del mismo contexto local de la nube. Como se ha visto con anterioridad, en el proceso iterativo de construcción de cada triángulo de la malla se realiza una evaluación de los posibles triángulos que forma un segmento con el resto de puntos. Gracias al KD-Tree, sólo se construyen y evalúan un número muy reducido de posibles triángulos; lo que evita

muchos cálculos. Cuanto mayor es la nube de puntos, más imprescindible es el uso de la estructura de datos KD-Tree.

3.2. Relleno de agujeros

Los datos obtenidos mediante el escaneo de modelos 3D normalmente contienen áreas sin información que pueden deberse a la propia geometría del objeto medido o a la existencia de sombras. Los modelos incompletos no son estéticamente atractivos. Además, en gráficos por ordenador surgen a menudo situaciones en las que sólo se puede trabajar con modelos completamente cerrados. De ahí la importancia de implementar un algoritmo que cierre los huecos existentes en la triangulación. El relleno de agujeros es un proceso similar al relleno progresivo (véase epígrafe 3.1.1), pero considerando únicamente los segmentos frontera, que son aquellos que se utilizaron una única vez.

3.3. Diezmado

El diezmado de una triangulación es el proceso por el cual se reduce el número de triángulos o caras del modelo mientras se mantiene la forma, volumen y límites de éste de una manera precisa (ajustándose a los requerimientos del trabajo que se está realizando). En este artículo se utiliza un algoritmo de diezmado de triángulos (Fig. 5), que consiste en seleccionar un triángulo a descartar, eliminar los tres vértices que lo componen, y volver a triangular el agujero resultante.

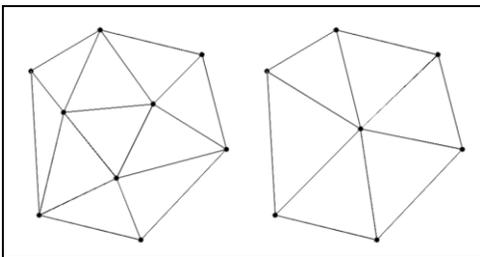


Fig. 5. Diezmado de la malla.

El proceso consiste en recorrer uno a uno los triángulos de la malla, evaluando si es posible eliminarlo. Esta evaluación se realiza calculando los ángulos entre las normales del triángulo central (el de estudio) y sus adyacentes; en caso de que alguno de esos ángulos supere cierto umbral definido por el usuario (por ejemplo, 20°), el triángulo central no será eliminado. Esto asegura que no se eliminen triángulos que formen ángulos entre normales grandes, es decir, triángulos que estén en una superficie que conforme una esquina, líneas de rotura, etc. De esta manera se preserva la forma del modelo diezmado, eliminando triángulos solamente en zonas más planas u homogéneas.

3.4. Optimización de procesos

Una estrategia para optimizar el tiempo de cálculo de los procesos que se han descrito en el apartado anterior es realizar un tratamiento previo de los datos. Para este propósito se optó por la realización de una transformación geométrica sólido-rígido, previa a la estructuración KD-Tree. Mediante esta

transformación, se alinea la nube de puntos al sistema de coordenadas, haciendo que sea óptima la subdivisión del espacio que realiza el KD-Tree. El fraccionamiento del espacio en la construcción del árbol KD-Tree se realiza de manera ortogonal al sistema coordenado, por tanto el proceso es más eficaz si la nube de puntos está también alineada de forma ortogonal. Esta operación cuenta con dos pasos.

1) *Análisis de Componentes Principales (ACP)*: Con esta operación se pretende buscar un nuevo sistema de coordenadas que se encuentre alineado geoméricamente con la nube de puntos. Por ejemplo, si una nube de puntos forma una figura alargada en una determinada dirección, uno de los tres ejes coordenados deberá mostrarse paralelo a esa dirección. Y lo mismo ocurre con los otros dos ejes de coordenadas. La manera de encontrar esa tendencia direccional de la nube de puntos es mediante la aplicación de un ACP [SMI02], que permita buscar una serie de componentes principales del conjunto de los datos espaciales. Estos componentes resultan ser vectores que indican precisamente las direcciones hacia las que tiende la nube de puntos. Si se escogen los tres primeros componentes principales del análisis, se obtienen los tres ejes coordenados del nuevo sistema, puesto que las componentes son ortogonales entre sí, como se puede apreciar en la Fig. 6.

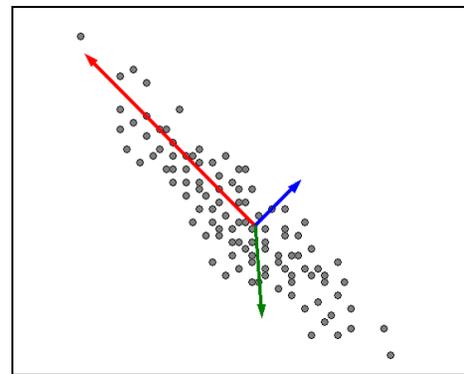


Fig. 6. Sistema de coordenadas local obtenido con el ACP.

2) *Rotaciones*: Una vez conocemos las direcciones del sistema de coordenadas local de la nube de puntos, se puede realizar la transformación sólido-rígido. De manera que se produce una rotación del conjunto de datos sin alterar su tamaño o su forma.

3.5. Comparación entre mallas

Los algoritmos de diezmado de mallas [ROY02] [ISE03] utilizan diferentes criterios para medir el error en la idoneidad de las superficies de aproximación. Entendido dicho error como la diferencia entre la forma de la malla original y la simplificada. Por lo general, estos algoritmos no devuelven las medidas del error introducido a la vez que

se simplifica la malla. Por lo tanto, una herramienta de comparación entre mallas es útil para caracterizar los algoritmos de simplificación de mallas. También es útil en la comparación de dos mallas cualesquiera que se superpongan en un sistema de coordenadas común a ambas. La herramienta desarrollada se basa en una cuantificación de las distancias de separación entre ambas mallas (Fig. 7). Tomando una de las dos mallas como superficie de referencia, las distancias a medir son aquellas que hay entre esta malla de referencia y la segunda malla, midiéndose la distancia a lo largo de la normal a la superficie de referencia.

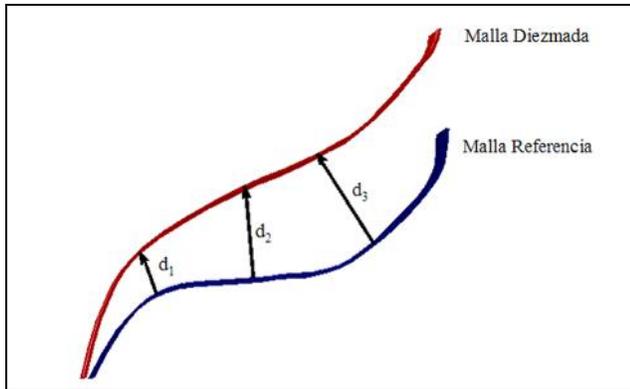


Fig. 7. Distancia entre mallas.

4. RESULTADOS EXPERIMENTALES

El programa ha sido ejecutado en un PC con sistema operativo Microsoft Windows XP Profesional, versión 2002 y Service Pack 2. Las características del hardware son las siguientes: Intel® Core™2 Quad CPU, Q8200 @ 2.33GHz., y 3.18 GB de RAM.

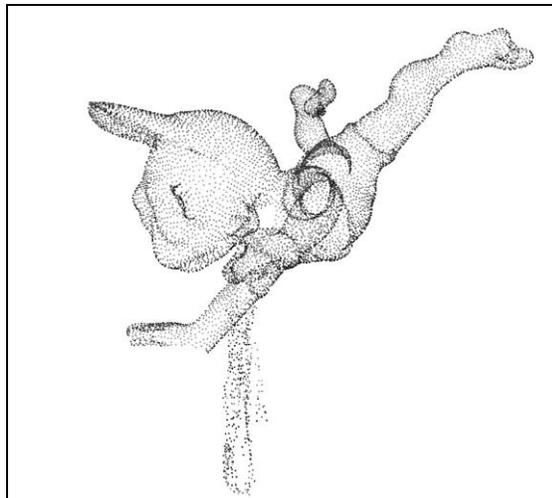
La nube de puntos empleada para probar el programa corresponde al levantamiento realizado mediante láser escáner de la Falla 2009 Sección Especial Almirante Cadarso – Conde Altea (Fig. 8). La captura de datos se realizó el día 16 de marzo de 2009, utilizando el láser escáner terrestre ScanStation 2 de Leica Geosystems (Fig. 8). La precisión nominal del instrumental láser en la medida de un punto es ± 6 mm a 50 m. Se realizaron barridos desde diez estaciones distintas. Para testear la aplicación se trabajó exclusivamente sobre una de las figuras de la falla también conocidas como Ninots. La Fig. 9 (a) muestra una imagen del ninot elegido de dimensiones 4.484 x 6.037 x 7.938 m. La nube de puntos contiene 13850 puntos procedente de 9 barridos. Los tratamientos que se ha realizado antes de comenzar con la triangulación son un filtrado de redundancias y un suavizado de puntos. El filtrado de redundancias es utilizado para reducir el número de puntos allá donde existe superposición; mientras que el suavizado de puntos proporciona una reducción de la aspereza de la nube de puntos, eliminando ruido sin alterar el total de puntos. La Fig. 9 (b) y (c) muestra la nube de puntos empleada y el modelo resultante.



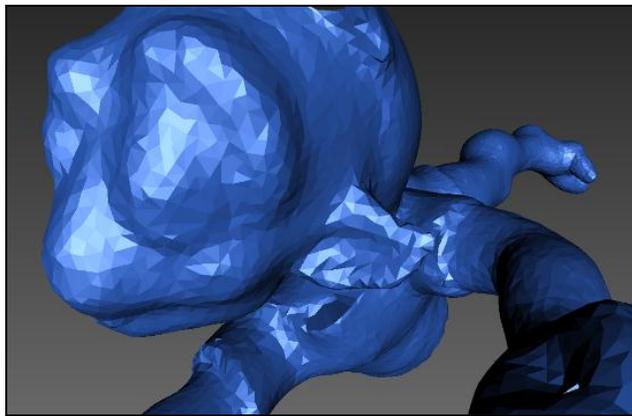
Fig. 8. Estacionamiento del láser escáner terrestre frente a la Falla 2009 Sección Especial Almirante Cadarso – Conde Altea.



(a)



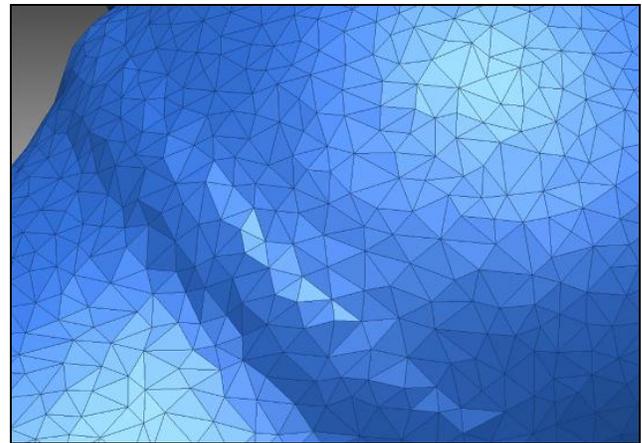
(b)



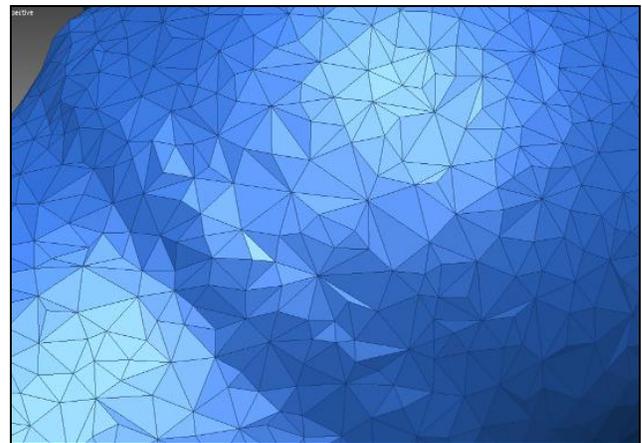
(c)

Fig. 9. Vistas en perspectiva del ninot: a) Fotografía natural; b) Nube de puntos; c) Modelo.

A continuación se presentan dos detalles comparativos entre este modelo original y el modelo diezmado (Fig. 10). En cuanto al tiempo de ejecución total (sumando los tiempos de los subprocesos de triangulación, relleno de agujeros y diezmado), sin el uso de la estructura KD-Tree es de 5 min 03 s; mientras que incluyendo dicha estructura, el tiempo es 2 min 37 s.



(a)



(b)

Fig. 10. Detalle del ninot: a) modelo original; b) modelo diezmado.

La Tabla 1 muestra el número total de puntos y triángulos de la malla antes y después del diezmado.

Malla Original		Malla Diezmada	
Nº puntos	Nº triángulos	Nº puntos	Nº triángulos
13850	29218	9567	20851

Tabla 1. Número de puntos y de triángulos de las mallas.

Por último, para comprobar las discrepancias entre la malla original y la diezmada, se presenta una superficie de comparación entre modelos, donde las discrepancias entre ellos se muestran a partir de un código de colores (Fig. 11). Los triángulos se colorean individualmente siguiendo la rampa de intervalos de color de la Tabla 2.

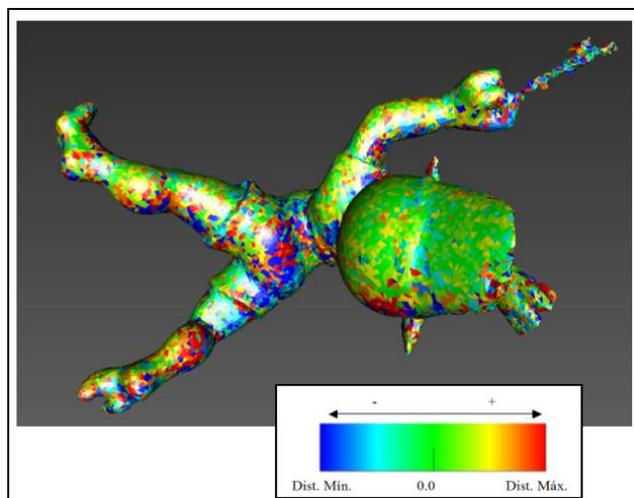


Fig. 11. Superficie de comparación entre modelos.

En la Tabla 2 se muestra de manera numérica los intervalos de color utilizados en la clasificación de distancias. Se ha hecho una clasificación con el mismo número de observables para cada intervalo.

Intervalo	Color	Distancias (m)
1	■	$-0.0040 < d \leq -0.0028$
2	■	$-0.0028 < d \leq -0.0016$
3	■	$-0.0016 < d \leq -0.0004$
4	■	$-0.0004 < d \leq 0.0008$
5	■	$0.0008 < d \leq 0.0019$
6	■	$0.0019 < d \leq 0.0031$
7	■	$0.0031 < d \leq 0.0043$
8	■	$0.0043 < d \leq 0.0055$
9	■	$0.0055 < d \leq 0.0067$

Tabla 2. Intervalos en la clasificación de distancias.

5. CONCLUSIONES Y RECOMENDACIONES

Este artículo presenta los desarrollos llevados a cabo para resolver ciertas operaciones complejas en la reconstrucción o modelado de nubes de puntos de objetos puramente 3D obtenidas con tecnología láser escáner terrestre. La herramienta es capaz de convertir las nubes de puntos densas en modelos tridimensionales formados por mallas, cuya unidad base es el triángulo. A su vez, los modelos pueden ser diezmados con el propósito de reducir el peso del modelo manteniendo el tamaño y la forma. Se presenta un algoritmo que reduce de manera inteligente el número de puntos de la nube y el número de triángulos de la malla. El proceso implica la creación de un nuevo modelo 3D diezmado. En la (Fig. 10) se puede apreciar que, pese a una reducción del número de triángulos y puntos en torno al 40% (Tabla 1), el modelo mantiene la forma. Es posible cuantificar y clasificar las diferencias entre el modelo original y el diezmado (Tabla 2).

Por otra parte, el uso de la estructura KD-Tree es fundamental, ya que implica una notable reducción del tiempo de ejecución del algoritmo, como se muestra en la Sección 4. Además, cuanto mayor es la nube de puntos, más imprescindible es el uso del KD-Tree, puesto que sin esta estructura el tiempo de ejecución se incrementa exponencialmente.

Es imprescindible una parametrización adecuada si pretendemos obtener resultados óptimos. Existe la posibilidad de elegir el valor de algunos parámetros de cálculo de las condiciones en la construcción, descritos en el apartado 3.3.1., como son: el umbral de radio, la tolerancia del KD-Tree, el umbral de radio del relleno progresivo, y el umbral de relleno de agujeros. Según el acierto al dar estos valores, se obtiene mayor o menor calidad en el resultado final. No obstante, la aplicación asigna a cada parámetro ciertos valores por defecto, obtenidos mediante ensayo y error, que se estiman generalmente buenos.

Los resultados obtenidos tras el proceso de diezmado y la comparación entre mallas (original y diezmada) con las nubes de puntos utilizadas, muestra que los errores básicamente están repartidos de una manera homogénea sobre la superficie del modelo (Fig. 11). Esto garantiza que, al diezmar la malla de un modelo, no se produzcan pérdidas en la forma y tamaño de éste, aunque lógicamente sí existe una tendencia a haber errores menores en zonas planas, puesto que se necesitan menos puntos para definir las que las superficies curvas.

Por último y como trabajo futuro, se contempla la posibilidad de continuar el desarrollo de nuevas operaciones de gestión de modelos 3D, como pueden ser algoritmos de limpieza de mallas (que encuentran y corrigen errores en la malla), de registro automático de mallas, y la adición de texturas al modelo.

AGRADECIMIENTOS

Los resultados presentados se enmarcan dentro del proyecto de investigación HAR2010-18620 del Ministerio de Ciencia y Tecnología.

REFERENCIAS

- [KOV07] James Kovacs, “C#.NET History Lesson”, 2007, <<http://jameskovacs.com/2007/09/07/cnet-history-lesson/>>, [13 septiembre 2012].
- [ROD01] Alberto Rodríguez, Reconstrucción 3D. “Generación de modelos tridimensionales basados en técnicas de triangulación y acoplamiento de caras: 3D en Internet”, 2001, <<http://www.elai.upm.es:8009/spain/Investiga/G>

- CII/personal/lrodriguez/lrodriguezE.htm>, [1 de septiembre 2012].
- [FIS07] Matthew Fisher, Boris Springborn, Peter Schröder, Alexander I. Bobenko, “An Algorithm for the Construction of Intrinsic Delaunay Triangulations with Applications to Digital Geometry Processing”, *Computing*, 82(2-3), 2007, pp. 199-213
- [HIL96] A. Hilton, A. J. Stoddart, J. Illingworth, T. Windeatt, “Reconstruction of 3D Delaunay Surface Models of Complex Objects”, *Systems, Man, and Cybernetics, IEEE International Conference*, vol. 4, 1996, pp. 2445-2450.
- [XU10] Li Xu, Yang Ke-biao, Yin Peng-ju, “Research on Triangular Mesh Generation Algorithm of Point Clouds”, *Computer Engineering and Technology (ICCT)*, 2010 2nd International Conference, vol. 4, 2010, pp. 638-641.
- [SHA02] Chandran Sharat, “Introduction to KD-Tree”, 2002, <<http://www.cs.umd.edu/class/spring2002/cmsc420-0401/pbasic.pdf>>, [7 de septiembre 2012].
- [SMI02] Lindsay I. Smith, “A tutorial on Principal Components Analysis”, 2002, <http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf>, pp. 1-26, [7 de septiembre 2012].
- [ROY02] Michael Roy, Sebti Foufou, Frederic Truchetet, “Mesh Comparison using Attribute Deviation Metric”, *International Journal of Image and Graphics (IJIG)*, 4 (1), 2004, pp. 127-140.
- [ISE03] Martin Isenburg, Peter Lindstrom, Stefan Gumhold, Jack Snoeyink, “Large Mesh Simplification using Processing Sequences”, *IEEE Visualization 2003*, pp. 465-472.
- [PAN01] Zhigeng Pan, Kun Zhou, Jiaoying Shi, “A new mesh simplification algorithm based on triangle collapses”, *Journal of Computer Science and Technology*, vol. 16, núm. 1, 2001, pp. 57-63.

Santiago Fabado Salvador: Ingeniero Técnico en Topografía (2008) e Ingeniero en Geodesia y Cartografía (2011) por la Universitat Politècnica de València (UPV). En Julio del 2010 se une a GIFLE como becario de colaboración; actualmente de especialización de 2º ciclo. Las líneas de interés que sigue son los gráficos 3D por computador, visualización 3D, y tareas de gestión de nubes de puntos y modelos 3D. Principalmente desarrolla soluciones para la plataforma .NET, así como para otros entornos como son los dispositivos móviles y la Web.

Miriam Cabrelles López: Ingeniera Técnica en Topografía e Ingeniera en Geodesia y Cartografía por la UPV. Desde el año 2007 forma parte de GIFLE, y en la actualidad es técnico superior de investigación. Sus actuales líneas de investigación son: Fotogrametría de objeto cercano: Rectificación de imágenes y calibración de cámaras; Análisis termográficos en edificación y restauración: Integración de termografía y

fotogrametría, fusión de imágenes; y Escaneado láser terrestre para la generación de modelos 3D texturizados.

Ana Elena Seguí Gil: Ingeniera Técnica en Topografía e Ingeniera en Geodesia y Cartografía por la UPV. Integrante de GIFLE desde el año 2006, como técnico de investigación. Cabe destacar sus habilidades en distintos entornos de programación, entre los que destacan: Plataforma .NET (Lenguaje C#); Entorno ECLIPSE (Java) y Matlab, y otros lenguajes como XML y KML. Sus principales líneas de investigación son el desarrollo de soluciones en entornos cartográficos para la gestión y visualización de información vía Web, y soluciones fotogramétricas para la automatización de procesos y creación de escenarios 3D.

José Luis Lerma García: Profesor Titular de Universidad en la UPV. Ingeniero Técnico en Topografía e Ingeniero en Geodesia y Cartografía. Director de GIFLE. Miembro de diversos comités científicos internacionales como CIPA y VSMM. Su investigación se centra en el desarrollo de mejoras geomáticas en el entorno de la documentación patrimonial a partir de sensores métricos y multiespectrales. Autor de siete libros de texto relacionados con la fotogrametría y el escaneado láser, presenta y publica de manera regular en foros internacionales.