

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** co-edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE** European Network).

<<http://www.ati.es/novatica/>>
 <<http://www.ati.es/reicis/>>
 <<http://www.cepis.org/upgrade>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC**, **RITSI** e **HispanLinux**, junto a la que participa en **ProInnova**.

Consejo Editorial

Ignacio Aguillo Sousa, Guillem Ainsina González, María José Escalona Cuaresma, Rafael Fernández Calvo (presidente del Consejo), Jaime Fernández Martínez, Luis Fernández Sanz, Didac Lopez Viñas, Celestino Martín Alonso, José Onofre Montesa Andrés, Francesc Noguera Puig, Ignacio Pérez Martínez, Andrés Pérez Payeras, Viktu Pons i Colomer, Juan Carlos Vigo López

Coordinación Editorial

Llorenç Pagés Casas <pages@ati.es>

Composición y autoedición

Jorge Lléser Gil de Ramales

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>

Administración

Tomas Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores

Acceso y recuperación de la información

José María Gómez Hidalgo (Optenet), <jmgomez@optenet.es>

Manuel J. María López (Universidad de Huelva), <manuel.mana@diehsia.uhu.es>

Administración Pública electrónica

Francisco López Crespo (MAE), <floc@ati.es>

Arquitecturas

Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>

Jordi Tabetlla Moragas (DAC-UPC), <jordit@ac.upc.es>

Auditoría SITIC

Marina Touriño Troitiño, <marinatourino@marinatourino.com>

Manuel Palao García-Suelto (ATI), <manuel@palao.com>

Derecho y tecnologías

Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV), <isabel.hernando@ehu.es>

Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

Enseñanza Universitaria de la Informática

Cristóbal Pareja Flores (DSIP-UCM), <cpajef@sis.ucm.es>

J. Ángel Velázquez Hurtado (ULSI, URJC), <angel.velazquez@urjc.es>

Entorno digital personal

Andrés Marín López (Univ. Carlos III), <amarin@iit.uc3m.es>

Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>

Estándares Web

Encarna Quesada Ruiz (Virati), <encarna.quesada@virati.com>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería), <jcarco@gmail.com>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young), <juan.baiget@ati.es>

Informática y Filosofía

José Ángel Olivás Varela (Escuela Superior de Informática, UCLM), <joseangel.olivas@uclm.es>

Roberto Feltrero Oreja (UNED), <rfeltrero@gmail.com>

Informática Gráfica

Miguel Chover Selles (Universitat Jaume I de Castellón), <mchover@lsi.uji.es>

Roberto Vivó Hernández (Enrographics, sección española), <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosin (ULSI-UPV), <dolado@lsi.ehu.es>

Daniel Rodríguez García (Universidad de Alcalá), <daniel.rodriguez@uah.es>

Inteligencia Artificial

Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV), <vbotti.vinglada@dsic.upv.es>

Interacción Persona-Computador

Pedro M. Latorre Andrés (Universidad de Zaragoza, AIPO), <platorre@unizar.es>

Francisco L. Gutiérrez Vela (Universidad de Granada, AIPO), <fgutierrez@ugr.es>

Lengua e Informática

M. del Carmen Ugarte García (ATI), <cuagate@ati.es>

Lenguajes Informáticos

Oscar Belmonte Fernández (Univ. Jaime I de Castellón), <bellem@lsi.uji.es>

Inmaculada Coma Taty (Univ. de Valencia), <inmaculada.coma@uv.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo), <xggo@uvigo.es>

Manuel Palomar (Univ. de Alicante), <mpalomar@dsi.ua.es>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI), <gmon.trotti@gmail.com>

Mike Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid), <mikelbno_uni@yahoo.es>

Profesión Informática

Rafael Fernández Calvo (ATI), <rfrcalvo@ati.es>

Miguel Sarrías Grijó (ATI), <msarrias@ati.es>

Redes y servicios telemáticos

José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>

Juan Carlos López López (UCLM), <juanarlos@uclm.es>

Robótica

José Cortés Arenas (Sopra Group), <jcortesa@gmail.com>

Juan González Gómez (Universidad Carlos III), <juan@isrobotics.com>

Seguridad

Javier Arellito Bertolin (Univ. de Deusto), <jarellito@deusto.es>

Javier López Muñoz (ETSI Informática-UMA), <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <faalonso@puente@dit.upm.es>

Software Libre

Jesus M. González Barahona (Universidad Politécnica de Madrid), <jm.gonzalez@upm.es>

Israel Herráiz Tabernerio (UAJ), <isra@herraz.org>

Tecnología de Objetos

Jesus García Molina (DIS-UM), <jmolina@um.es>

Gustavo Rossi (LFIA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>

Tecnologías para la Educación

Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>

César Pablo Córcoles Briogio (UOC), <ccorcoles@uoc.edu>

Tecnologías y Empresa

Didac López Vilas (Universitat de Girona), <didac.lopez@ati.es>

Francisco Javier Gomis Sánchez (Indra Sistemas), <fgomis@indrasistemas.com>

Tendencias tecnológicas

Alonso Álvarez García (TID), <aalvarez@tid.es>

Gabriel Marín Fuentes (Interbits), <gabim@atnet.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga), <aguayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º dcha., 28006 Madrid

Tlf: 91 4029391; fax: 91 3093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Reino de Valencia 23, 46005 Valencia

Tlf: fax 963330392 <secreval@ati.es>

Administración y Redacción ATI Cataluña

Via Laietana 46, ppal. 1º, 08003 Barcelona

Tlf: 934125236; fax 934127713 <secregen@ati.es>

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza.

Tlf: fax 976235181 <secreara@ati.es>

Redacción ATI Andalucía

Redacción ATI Galicia <secregal@ati.es>

Suscripción y Ventas

<<http://www.ati.es/novatica/interres.html>>, ATI Cataluña, ATI Madrid

Publicidad

Padilla 66, 3º dcha., 28006 Madrid

Tlf: 91 4029391; fax: 91 3093685 <novatica@ati.es>

Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona

Depósito legal: B 16.154-1975 - ISSN: 0211-2124; CODEN NOVAEC

Portada: La escarga luminosa - Ochoa, Arias Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

editorial

Implicaciones de las TIC verdes > 02

noticias de IFIP

Resumen de la Asamblea General de IFIP 2011 > 03

Ramón Puigjaner Trepal

Reunión anual del TC-2 (Software – Theory and Practice) > 06

Antonio Vallecillo Moreno

actividades de ATI

Entrega de la V Edición del Premio Novática > 03

en resumen

Informática y conservación del planeta > 10

Llorenç Pagés Casas

monografía

TIC verdes: Tendencias y retos

(En colaboración con **UPGRADE**)

Editores invitados: Juan Carlos López López, Giovanna Sissa, Lasse Natvig

Presentación. TIC verdes: El compromiso de la Sociedad

de la Información con un desarrollo sostenible > 07

Juan Carlos López López, Giovanna Sissa, Lasse Natvig

Encuesta de CEPIS sobre TIC verdes – Examinando la conciencia sobre

TIC verdes en las organizaciones: Hallazgos iniciales > 11

Carol-Ann Kogelman en nombre del Green ICT Task Force de CEPIS

Los cinco temas más descuidados en las TI verdes > 15

Lorenz M. Hilty, Wolfgang Lohmann

Servicios de computación: oportunidades "verdes" y riesgos > 19

Giovanna Sissa

El software bueno, el bello y el malo – en busca de los factores verdes

de calidad de software > 24

Juha Taina

Técnicas de Inteligencia Artificial aplicadas a la Red Eléctrica Inteligente (Smart Grid) > 29

María José Santolimia Romero, Xavier del Toro García, Juan Carlos López López

Computación verde: economizando energía a través de la regulación,

la simplicidad y la paralelización > 35

Lasse Natvig, Alexandru Ciprian Iordan

Hacia un Cloud Computing sostenible en Europa > 42

Kien Le, Thu D. Nguyen, Iñigo Goiri, Ricardo Bianchini, Jordi Guitart Fernández, Jordi Torres Viñals

Eficiencia energética en centros de proceso de datos:

Investigación y realidad práctica > 47

Marina Zapater Sancho, Patricia Arroba García, José Manuel Moya Fernández, Zorana Bankovic

secciones técnicas

Informática Gráfica

Modelos de Papel 3D Personalizados: papermodels.uji.es > 52

José Ribelles Miguel, Mª Ángeles López Malo

Interacción Persona-Computador

ObservAR, sistema de realidad aumentada multiusuario para exposiciones > 57

Jesus Gimeno, Ricardo Olanda, Bibiana Martinez, Fernando M. Sanchez

Tecnologías y empresa

Impactos derivados de la adhesión a un código de conducta en materia

de contratación electrónica > 61

David López Jiménez, María Dolores Gallego Pereira, Salvador Bueno Ávila

Referencias autorizadas > 66

sociedad de la información

Programar es crear

El problema de la función exponencial

(Competencia UTN-FRC 2010, problema A, enunciado) > 74

Julio Javier Castillo, Diego Javier Serrano

El problema del decodificador

(Competencia UTN-FRC 2010, problema C, solución) > 75

Julio Javier Castillo, Diego Javier Serrano

asuntos interiores

Coordinación editorial / Programación de Novática / Socios Institucionales > 77

Tema del próximo número: "Gestión de riesgos"

José Ribelles Miguel, M^a
Angeles López Malo

*Institute of New Imaging Technologies,
Universitat Jaume I (Castellón)*

<{ribelles,lopeza}@uji.es>

Modelos de Papel 3D Personalizados: papermodels.uji.es

1. Introducción

El sector del entretenimiento en la industria es uno de los principales campos en cuyo crecimiento la ingeniería de la computación y en especial el área de la informática gráfica juegan un importante papel. Tradicionalmente, el uso de los gráficos generados por computador ha estado muy ligado al segmento del videojuego, ya sea para su uso en ordenadores personales como para consolas, y a los del cine y la televisión, mediante personajes virtuales, efectos especiales, o series de dibujos animados generados íntegramente por ordenador.

Probablemente, uno de los entretenimientos más clásicos de nuestras vidas sea el recortable de papel. Recortables de muñecas en 2D, o modelos geométricos simples en 3D como el cubo o el cilindro son ejemplos que habitualmente han aparecido durante la edad escolar. Y cómo no, el arte milenario de la papiroflexia y todas sus variantes que permiten crear innumerables formas de seres u objetos. En la revolución tecnológica en la que nos encontramos y, en concreto, los avances producidos en diversos aspectos de la informática gráfica en la última década permiten que la aplicación de los modelos de papel ofrezca desafíos de mayor complejidad y acabados mucho más versátiles. Así, resulta fácil encontrar en Internet multitud de modelos 3D a escala, desplegados y dispuestos en hojas de papel listos para ser contruidos, incluyendo, entre otros, construcciones emblemáticas (como catedrales o palacios), naves (ya sean civiles, militares o espaciales), vehículos, animales y personajes de ficción. En definitiva, se puede afirmar que el papel sigue siendo hoy en día un elemento básico en el entretenimiento tanto de pequeños como de mayores.

Una característica que se ha incorporado al mundo del comercio es la posibilidad de personalizar un producto a gusto del consumidor. En general, esta personalización supone adecuar el proceso de fabricación en uno o varios puntos en los que se utilizan los datos proporcionados por el propio usuario. Hoy en día, es posible obtener tazas, alfombrillas para el ratón, carteras, portavasos, toallas y, en definitiva, una larga lista de productos, personalizados generalmente con fotografías proporcionadas por el propio cliente. En el campo concreto del entretenimiento, como por ejemplo son los puzzles, se explota esta capacidad con gran éxito y así es

Resumen: Una característica que se ha incorporado al mundo del comercio hace ya algunos años es la posibilidad de personalizar un producto a gusto del consumidor. También en el sector del entretenimiento se ha aplicado con éxito, como es el caso de los puzzles. Gracias al desarrollo de diferentes tecnologías, pertenecientes principalmente al campo de la informática gráfica, esta personalización se puede aplicar en uno de los entretenimientos más clásicos de nuestras vidas, el recortable de papel. El objetivo de este artículo es mostrar cómo combinar estas tecnologías, describiéndolas y justificando su aplicación, para proporcionar modelos de papel 3D personalizados, asegurando una calidad muy alta en el acabado final del producto. También se presenta un prototipo de estructura cliente-servidor que cuenta con una interfaz intuitiva y que permite a cualquier usuario elegir, personalizar y recibir el producto de forma sencilla y rápida.

Palabras clave: Desplegado, modelos de papel, remallado, simplificación.

posible adquirir un puzzle donde la imagen a reconstruir la proporcione el propio usuario y además sea éste el que especifique tamaño y número de piezas. Sin embargo, este tipo de producto requiere de una maquinaria especializada lo que al mismo tiempo hace que se limite la personalización a una serie de opciones preestablecidas por el fabricante con el fin de que el precio final del producto sea competitivo comparado con el mismo producto sin personalizar.

Este artículo presenta la combinación de la creación de modelos de papel 3D con la personalización de su acabado. Las siguientes circunstancias van a influir directamente en que esta combinación pueda realizarse con éxito:

- Hay disponibles en Internet de forma gratuita una amplia variedad de objetos 3D modelados mediante triángulos.
- La materia prima de construcción es el papel, que resulta fácil de obtener y de manipular, barato y, muy importante, reciclable.
- La materialización del producto se reduce a simplemente la impresión del recortable en hojas de papel mediante una impresora, dispositivo del que se dispone con facilidad bien en casa o bien a través de una copistería a precios muy asequibles.
- Y por último, el hardware gráfico actual permite que en tiempo real se puedan aplicar opciones de personalización al mismo tiempo que se asegura una calidad muy alta en el acabado del producto.

El objetivo de este artículo es mostrar cómo diferentes tecnologías, pertenecientes principalmente al campo de la informática gráfica, se pueden combinar para crear una aplicación novedosa como es la creación de modelos de papel 3D personalizados. En el artículo se

describen estas tecnologías al mismo tiempo que se justifica su aplicación. También se detalla el proceso de despliegue del modelo y las técnicas utilizadas para que en su dibujado se consiga una alta calidad en el producto final. Por último, se presenta el prototipo de estructura cliente-servidor que se ha desarrollado y que permite al usuario a través de una interfaz muy sencilla seleccionar un modelo 3D con un número de piezas adecuado al nivel de su habilidad así como de entre diversas opciones de acabado como color de superficie, color y grosor de las aristas o inclusión de fotografías que se utilizarán a modo de adhesivos o pegatinas. El programa cliente permite que el usuario observe el resultado antes de solicitar la petición de creación del recortable al servidor, mientras que éste la procesará en línea haciendo que el usuario obtenga el recortable instantes después de haberlo solicitado.

2. Visión general

El elemento central de esta aplicación es el modelo 3D. Hoy en día es posible conseguir estos modelos de forma gratuita o a un coste muy bajo en Internet. Muchos sitios web ofrecen estos productos clasificados en categorías como vehículos, personajes de ficción, animales, etc. Estos objetos se crean mediante herramientas de modelado 3D o bien utilizando escáneres y la primitiva geométrica de representación por excelencia es el triángulo. El proceso de creación de un recortable de papel comienza una vez obtenido el modelo geométrico 3D y consta de las siguientes etapas (ver **figura 1**):

1) Adecuación: El objetivo de esta etapa es modificar el objeto 3D original para, por una parte, conseguir que el número de triángulos del objeto sea adecuado a la experiencia y

habilidad del modelista, al mismo tiempo que proporcionar modelos de complejidad escalonada y así satisfacer a una amplia diversidad de público. Por otra parte, es importante también obtener una alta calidad en su aspecto, la cual viene condicionada principalmente por la calidad del triángulo, ya que este va a ser perfectamente visible una vez construido el recortable de papel.

2) Procesamiento: En esta etapa se produce el despliegado del modelo 3D, lo que implica dividir el modelo en piezas, cada una formada por un conjunto de triángulos, de manera que al llevarlas al plano no se produzcan intersecciones entre los triángulos que componen una misma pieza, ubicar las pestañas necesarias para su construcción y distribuir las piezas en un espacio de tamaño limitado dado por el tipo de hoja de papel.

3) Dibujado. Tras las dos etapas anteriores el recortable está listo para ser generado. Falta que el usuario realice la petición especificando los datos de acabado que harán del recortable un producto personalizado. Entonces, el sistema deberá dibujar el producto solicitado y entregarle un archivo *pdf* listo para imprimir. Como el objetivo es que el usuario obtenga el resultado en tiempo real, se utiliza el estándar gráfico *OpenGL* que permite aprovechar al máximo el rendimiento de los procesadores gráficos actuales, dando una rápida respuesta a este proceso, al mismo tiempo que se asegura una alta calidad en el producto final.

3. Adecuación

Aunque se pueden encontrar objetos 3D modelados con unas pocas decenas de

polígonos, lo habitual es encontrar objetos que contienen del orden de miles e incluso del orden de cientos de miles de triángulos. Influye, aparte de la propia complejidad del objeto y del método de adquisición utilizado, el hecho de que la representación de superficies curvas se realiza mediante una aproximación de caras planas. Además, hay que tener en cuenta que lo habitual es que estos objetos se creen con el único fin de ser visualizados en escenarios tridimensionales, que también el hardware gráfico de consumo disponible en ordenadores de bajo coste puede visualizar grandes cantidades de polígonos manteniendo tasas interactivas (alrededor de 75 imágenes por segundo), y que por otra parte el esfuerzo del modelador es mucho mayor cuando se le exige que el modelo tenga calidad al mismo tiempo que un número pequeño de caras. Todos estos motivos hacen que en general el número de triángulos de los modelos sea demasiado elevado, incluso para modelistas experimentados, y que resulte muy difícil encontrar modelos compuestos por un número pequeño de polígonos manteniendo al mismo tiempo una alta calidad visual, lo cual es muy importante en la aplicación caso de estudio.

Así, el primer problema que aparece es cómo reducir esta cantidad de triángulos a una cantidad que sea adecuada a la habilidad y capacidad de cualquier aficionado. Este problema, motivado por otras aplicaciones, ha sido ampliamente estudiado principalmente desde finales de los años 90 y es conocido con el nombre de simplificación. Se han presentado numerosos métodos y también hay trabajos que realizan una revisión del estado del arte con clasificaciones y comparativas [1][2][3]. En cualquier caso, los métodos de simplificación nos abren dos posibles caminos. El primero consiste en ofrecer al usuario, de forma análoga al mundo de los puzzles, un conjunto precalculado de aproximaciones del mismo modelo original pero diferentes en cuanto al número de triángulos que la forman y así, por ejemplo, ofrecer recortables de modelos compuestos por 50, 100, 250, 500 y 1.000 triángulos. La segunda posibilidad consiste en ofrecer un rango de aproximaciones casi continuo permitiendo que el usuario, en tiempo real, pueda elegir un modelo compuesto por prácticamente el número de triángulos que desee. Para llevar a la práctica esta segunda posibilidad, es necesario utilizar otra técnica también muy estudiada que es el modelado multirresolución [4][5]. Un esquema de representación multirresolución permite almacenar y visualizar de forma eficiente el resultado de la simplificación de un objeto, siendo capaz de reproducir todas las situaciones que el método de simplificación puede generar. La construcción del modelo multirresolución es un proceso que se realiza una vez [6], fuera de línea, mientras que en tiempo de ejecución la aplicación reconstruye

una aproximación del objeto con el nivel de detalle o complejidad seleccionado por el usuario a través de la interfaz de la aplicación, en la que utilizando un control como por ejemplo una barra de desplazamiento, hace variar el número de triángulos que lo componen al mismo tiempo que lo está observando. El uso de este tipo de representación no supone ninguna carga adicional para el sistema ni tampoco requiere un hardware de características especiales como se ha demostrado en la literatura [6], por lo que resultan muy adecuadas para la aplicación que nos ocupa.

Sin embargo, la mayor parte de los métodos de simplificación se han pensado para que su resultado se visualice en un mundo 3D, por lo que se preocupan sobre todo de preservar la geometría y la topología del modelo. Pero ahora el objetivo es obtener un recortable y construirlo en papel, lo que añade una diferencia muy importante y es lo que se conoce con el nombre de calidad del triángulo. Mientras que, en las aplicaciones donde la malla de triángulos resultante del proceso de simplificación se va a visualizar no importa la calidad de los triángulos generados (ya que las técnicas de iluminación y sombreado nos van a disimular y ocultar en gran medida la geometría que realmente se está utilizando), en las aplicaciones donde se va a construir con papel el aspecto de los triángulos es perfectamente perceptible para cualquier usuario, influyendo decisivamente en el aspecto final del recortable. Aunque sí que existen algunos métodos de simplificación que, además de preservar la forma tratando de mantener las aristas importantes o, en general, los detalles característicos propios del objeto, también se han preocupado de tener en cuenta propiedades de material y así no mezclar en una operación de simplificación triángulos que posean propiedades de material distintas como, por ejemplo, texturas diferentes, prácticamente ninguno se ha preocupado de que la malla de triángulos obtenida mantenga el aspecto regular que muy probablemente muestra el original y que se desea apreciar también en el modelo objeto del recortable final.

Por lo tanto, el segundo problema que aparece es cómo conseguir una malla con triángulos de calidad. La calidad de un triángulo viene dada por su relación de aspecto. Por ejemplo, un triángulo equilátero maximiza su relación de aspecto. Para conseguirlo se pueden utilizar métodos de remallado [7][14] ya que estos métodos mejoran considerablemente la calidad de los polígonos que representan el modelo. Dada una malla 3D, el objetivo es calcular otra malla que aproxime a la inicial y al mismo tiempo satisfaga algunos requisitos de calidad como número de vértices, regularidad, tamaño, etc. Estos métodos son piezas clave en aplicaciones de modelado, edición, animación y simulación

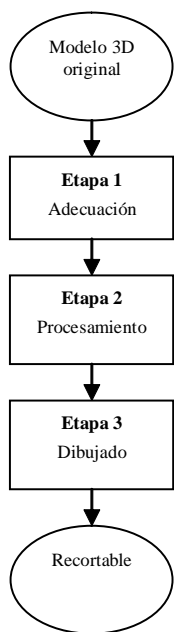


Figura 1. Secuencia de operaciones del proceso de creación del recortable.

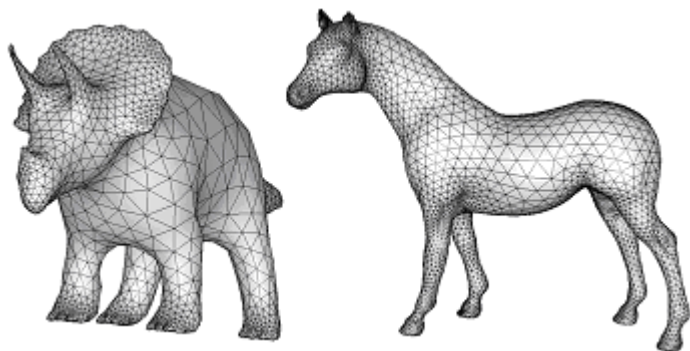


Figura 2. Resultados de un remallado muy regular. Imagen cortesía de Surazhsky y Gotsman [8].

entre otras, y por este motivo ha recibido mucha atención en el pasado. Desde el punto de vista de la aplicación que nos ocupa, prácticamente cualquier método de remallado va a producir una malla agradable a la vista. Sin embargo, hay que destacar el método propuesto por Surazhsky y Gotsman [8] que de forma concreta se ha preocupado por conseguir una malla muy regular enfatizando su preocupación por la calidad del triángulo (ver figura 2). Por contra, los objetos remallados con este método constan todavía de unos cuantos miles de triángulos, lo que puede ser válido para modelistas de mucha experiencia pero no para el público en general. Un método que sí es válido para generar objetos formados por menos triángulos es el presentado por Dong et al. [9]. Aunque visualmente no produce una malla tan estéticamente agradable como la anterior, es un método fácil y sencillo de implementar, rápido en ejecución, con el que se puede conseguir una malla poligonal muy regular, de alta calidad, formada por triángulos y cuadriláteros con buena relación de aspecto y el número de polígonos deseado (ver figura 3).

4. Procesamiento

Una vez conseguido el modelo 3D con el número de triángulos deseado, hay que realizar su despliegue como paso previo a su

construcción. El interés en construir modelos de papel por otros motivos distintos que como simple hobby también aparece en la industria principalmente en el área de la fabricación asistida por ordenador. Esto ha hecho que en la literatura se hayan propuesto diferentes soluciones a este problema [10][11][12]. Básicamente, estos métodos proponen calcular un conjunto de trozos que pueden ser desplegados en un plano, teniendo en cuenta que el material a utilizar es papel, lo que impone ciertas limitaciones físicas que otros materiales no imponen, y que todos juntos aproximan al modelo 3D con un cierto error. El objetivo común es conseguir que el número total de piezas sea pequeño, por supuesto muy por debajo del número de triángulos del modelo original, para así facilitar y acelerar las tareas de recortado y pegado de las piezas. Sin embargo, el resultado obtenido elimina la calidad visual que se había conseguido con los métodos de remallado debido a que las fronteras entre piezas son claramente perceptibles.

Por este motivo se propone simplemente desplegar la malla poligonal con un sencillo algoritmo de manera que el usuario construya exactamente lo que observa. Este algoritmo comienza a partir de una pieza formada por un único triángulo semilla e intenta des-



Figura 3. Comparación entre remallado a la izquierda y simplificado a la derecha. Imagen cortesía de Dong et al. [9].

plegar sus caras adyacentes haciendo crecer así la pieza. A su vez, cada cara adyacente intenta desplegar su otras dos caras adyacentes, siempre comprobando que no se produce intersección entre el triángulo que se despliega y el resto de triángulos ya añadidos a la pieza. Si se detecta intersección, dicho triángulo no puede ser añadido a la pieza en expansión y pasa a ser semilla de una nueva pieza. El algoritmo también comprueba si una cara adyacente ha sido ya añadida a otra pieza antes de intentar su despliegue. Los posibles triángulos a desplegar se introducen en una estructura de tipo pila o cola. Los resultados experimentales nos muestran que hay diferencia entre una estructura y otra. Si se usa una pila, el despliegado produce piezas en general de forma alargada, mientras que si se usa una cola se producen piezas de aspecto más agrupado. Al mismo tiempo, el crecimiento de la pieza se puede condicionar a no superar un número de triángulos dado o algún otro requisito como tamaño o límite del área de desplegado. También hemos comprobado que resulta conveniente iniciar el proceso de despliegue utilizando varios triángulos semilla, tratando de que estén lo más separadas posible en la superficie del modelo, y hacer crecer cada pieza en un triángulo de forma alternada. Los experimentos muestran que de esta manera se consiguen piezas de formas y aspectos más similares que comenzando con una única semilla y expandiendo la misma pieza mientras sea posible. Además, al mismo tiempo que se crean las piezas, se marcan las aristas de los triángulos donde añadir una solapa para poder realizar el pegado. Por ejemplo, para una esfera de 80 caras, utilizando dos semillas iniciales, se generan las dos piezas que se muestran en la figura 4.

Ahora es el momento de colocar las piezas en hojas de papel. Este proceso es conocido por su término en inglés, *nesting*. Si se desea automatizar por completo la creación del recortable, es necesario implementar un algo-



Figura 4. Resultado del proceso de despliegue para una esfera de 80 triángulos a partir de dos triángulos semillas.



Figura 5. Resultado del proceso de despliegue incluyendo los datos de personalización especificados por el usuario.

ritmo de *nesting*. Muchas empresas utilizan algoritmos de *nesting* para minimizar la cantidad de material desechado. Además, los algoritmos de *nesting* suelen tener en cuenta limitaciones del propio mecanismo de corte así como del material que se va a cortar. Sin embargo, trabajar con papel tiene la ventaja de no imponer restricciones de este tipo, ya que el modelista con sus tijeras será capaz de recortar cualquier pieza independientemente de cómo esté dispuesta. Por otra parte, al ser el papel barato y reciclable, el problema no es comparable al de las industrias del metal o las de la ropa, por ejemplo, donde la materia prima es muy cara y en muchos casos sus desechos no son reciclables. Cabe señalar también que las piezas generadas en el proceso de despliegue suelen ser muy irregulares lo que complica enormemente realizar un *nesting* automático y eficiente [15][16]. En el caso de que la aplicación ofrezca una serie de modelos precalculados, este proceso puede ser realizado a mano por un usuario experto. Aunque sea lento, esta tarea se realizará una única vez.

5. Dibujado

El modelo ya desplegado contiene la información necesaria para que se pueda dibujar a

falta únicamente de que el usuario especifique las condiciones que fijan el acabado del recortable. Si las opciones de acabado solo incluyesen color de superficie y color de arista se podría dibujar el recortable generando directamente un fichero *postscript* o *pdf*. Pero la posibilidad de añadir imágenes a modo de adhesivos o pegatinas (como el de la figura 5), u otros acabados más complejos, nos obliga a utilizar algún estándar gráfico como *OpenGL* y así utilizar el hardware gráfico para obtener en tiempo real la mayor calidad posible. Los primeros aceleradores gráficos para ordenadores personales surgieron al final de los años 90 y ya entonces eran capaces de dibujar miles de polígonos con textura por segundo. Pero para conseguir, además de velocidad, una excelente calidad en el producto final es necesario resolver el problema del efecto escalera, más conocido por su término anglosajón, *aliasing*. Este problema suele observarse sin dificultad en las aristas dibujadas. Un método que resuelve este problema y además se implementa en una GPU de hoy en día es el *multisampling*, es decir, obtener para cada píxel más de un valor de color y calcular su color definitivo mediante una operación. Además, otro beneficio de esta técnica es que

nos permite generar imágenes de resolución relativamente baja con una calidad muy satisfactoria. Así, por ejemplo, para una imagen del tamaño de una hoja de papel A4 hemos comprobado que una resolución de 1.400x2.000 píxeles proporciona calidad suficiente. Sin embargo, a mayor tamaño del papel, proporcionalmente hará falta mayor resolución en la imagen, por lo que fácilmente habrá que generar imágenes de resolución mayor de lo que la GPU es capaz. Este problema es aún mayor al utilizar la técnica de *multisampling* debido al consumo de memoria que conlleva. De esta manera, con el fin de poder generar una imagen de cualquier resolución es necesario dividirla en trozos y computar cada trozo por separado para juntarlos posteriormente obteniendo así la imagen completa. Por ejemplo, en las pruebas experimentales realizadas utilizando una GPU Nvidia Quadro FX 580, hemos dividido la imagen de 1.400x2.000 píxeles en 4 trozos cuando se utilizan 32 muestras por píxel.

La posibilidad de que el usuario proporcione una imagen para ser utilizada como textura también introduce una serie de problemas. Por ejemplo, el tamaño máximo de textura es una limitación que cada GPU establece, de 8.192 x 8.192 píxeles en nuestro caso, lo que obliga a realizar una operación de escalado en el caso de que el usuario proporcione una imagen que supere dicho límite. También influye mucho en la calidad final del producto la realización de algún tipo de operación de filtrado a la hora de determinar el color definitivo de cada píxel de la imagen final. Ya los primeros aceleradores gráficos con soporte para *OpenGL* incorporaban esta posibilidad realizando al menos un filtrado lineal. Este tipo de filtro proporciona calidad suficiente, pero hoy en día, gracias a las nuevas GPU

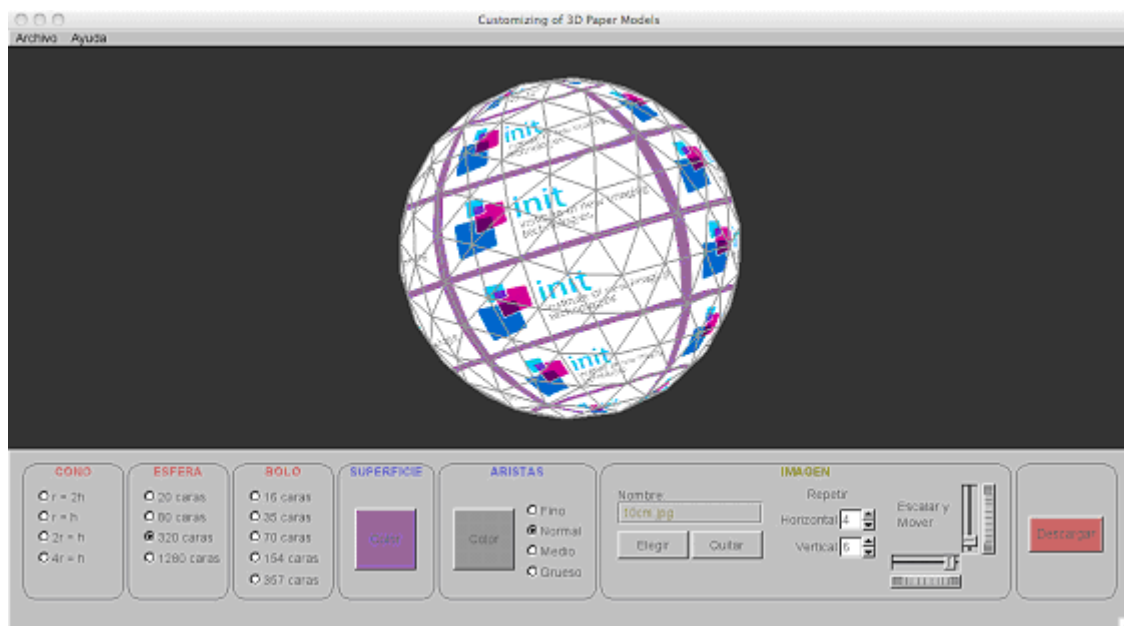


Figura 6. Interfaz del prototipo.

programables, es posible implementar y ejecutar en tiempo real métodos de filtrado más complejos que proporcionan un acabado de mayor calidad. Otro problema importante que aparece es la distorsión de la textura cuando se aplica en superficies poligonales formadas por cuadriláteros [17]. Este problema se aprecia con más claridad cuanto mayor sea la diferencia entre las proporciones de la textura y la del cuadrilátero sobre el que se aplica. Una solución simple que permite disminuir el problema hasta el punto de que no sea apreciable es subdividir cada cuadrilátero [18]. Esta subdivisión aumenta considerablemente el número de triángulos a dibujar, pero los procesadores gráficos más modernos son capaces de realizar esta subdivisión ellos mismos en tiempo de ejecución, lo que supone no sólo resolver el problema sino que además no haya una ralentización apreciable en el proceso de dibujado.

6. El prototipo

Hemos desarrollado un prototipo con estructura cliente-servidor, que se encuentra disponible en <<http://papermodels.uji.es/>>. El principal motivo que nos ha llevado a decidir este tipo de estructura reside en la tercera etapa del proceso, la de dibujado. Al mismo tiempo que el hardware gráfico da soporte a técnicas de acabado de alta calidad en tiempo real, también es cierto que no cualquier GPU es capaz de proporcionarlas. Mientras que para simplemente visualizar el modelo 3D coloreado, con o sin una textura, cualquier GPU es capaz de realizarlo a una tasa de imágenes interactiva, las tareas de *antialiasing*, filtrado de la textura de alta calidad o subdivisión de la malla requieren de una GPU más moderna y de mayores prestaciones. De esta manera, la etapa de dibujado se realiza íntegramente en el servidor, consiguiendo por tanto no limitar el uso de esta aplicación únicamente a usuarios que tengan acceso a ese hardware, normalmente más caro y disponible sólo en equipos de altas prestaciones.

El programa *cliente* permite al usuario seleccionar el objeto 3D así como las opciones de acabado. Al mismo tiempo lo puede observar con calidad suficiente como para tomar la decisión de cuándo solicitar el recortable. La **figura 6** muestra la interfaz de usuario correspondiente al cliente. En el prototipo se han incluido las siguientes opciones de personalización:

- Modelos: cono, esfera y bolo; donde para cada uno de ellos se ofrecen diferentes complejidades.
- Color de superficie y de aristas, seleccionables a través de sendos selectores de color.
- Grosor de aristas: el usuario dispone de cuatro posibilidades diferentes.
- Imagen: posibilidad de proporcionar una imagen lo que habilita a su vez diferentes opciones como aplicarla de forma repetida en direc-

ción horizontal o vertical, modificar su tamaño o desplazarla de su ubicación por defecto.

Cuando finalmente el usuario solicita la creación del recortable, el tipo de objeto seleccionado junto con los datos de acabado, incluyendo la imagen proporcionada por él mismo si así lo ha hecho, se envían al *servidor*. Este recibe los datos y procesa la petición de la siguiente manera. En primer lugar el servidor comprueba si la imagen recibida cumple con los límites de tamaño establecidos por la GPU y, en caso de superarse, se escala.

Además la imagen se transforma a formato RGB para poder ser utilizada con OpenGL y se le aplica un borde cuyo tamaño depende de los valores de personalización establecidos por el usuario. Para realizar estas tres operaciones se utiliza la aplicación *Convert* de Image Magick [13] que permite realizarlas en una única ejecución de la aplicación. En segundo lugar se dibuja el recortable utilizando toda la potencia de la GPU instalada en el servidor. Además, con *OpenGL* se puede redirigir la salida del sistema gráfico a un buffer proporcionado por la aplicación en lugar de a una ventana, técnica conocida con el nombre de dibujado *off-screen*. Esto permite que el servidor pueda generar las imágenes del recortable sin que haya una sesión iniciada y abierta, y que tampoco dependa de la resolución del modo gráfico establecido. Una vez obtenidas las imágenes, una imagen por página, el servidor también es el encargado de incluirlas en un nuevo archivo *pdf* y remitirlo al usuario. De esta manera también se libera de esta tarea al programa *cliente* que únicamente se limita a esperar durante unos segundos la recepción del archivo *pdf* para almacenarlo en el sistema de ficheros.

7. Conclusiones

En este artículo se ha mostrado cómo diferentes tecnologías, pertenecientes principalmente al campo de la informática gráfica, se pueden combinar para crear una aplicación novedosa como es la creación de modelos de papel 3D personalizados. El proceso de creación de un recortable de papel se ha dividido en tres etapas. En la primera etapa, adecuación, se han revisado las técnicas de simplificación, modelado multiresolución y remallado para conseguir que el objeto 3D tenga el número de triángulos adecuado a la experiencia y habilidad del modelista así como una alta calidad en sus aspectos. En la segunda etapa, procesamiento, se ha explicado como realizar el despliegue del modelo, la ubicación de las pestañas y la distribución de las piezas en hojas de papel. En la tercera y última etapa, dibujado, se han revisado las técnicas que van a permitir obtener el producto final de forma rápida al mismo tiempo que se asegura una alta calidad en el acabado. Por último, se ha presentado un prototipo de estructura cliente-servidor que permite al usuario a tra-

vés de una interfaz muy intuitiva elegir, personalizar y recibir el producto de forma sencilla y rápida.

Referencias

- [1] P. Cignoni, C. Montani, R. Scopigno. A comparison of mesh simplification algorithms. *Computers and Graphics*, 22(1), pág. 37-54, 1998.
- [2] P.S. Heckbert, M. Garland. Survey of polygonal surface simplification algorithms. *SIGGRAPH'97 Course Notes*, 1997.
- [3] V. Surazhsky, C. Gotsman. *A qualitative comparison of some mesh simplification software packages*. Technical report, 2005.
- [4] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, R. Huebner. *Level of detail for 3D Graphics*. Ed. Morgan Kaufmann, 2002.
- [5] J. Ribelles, A. López, O. Belmonte, I. Remolar, M. Chover. Multiresolution modeling of arbitrary polygonal surfaces: a characterization. *Computers and Graphics*, 26(3), pág. 449-462, 2002.
- [6] J. Ribelles, A. López, O. Belmonte. An Improved Discrete Level of Detail Model Through an Incremental Representation. *Proc. of Theory and Practice of Computer Graphics*, pág. 59-66, 2010.
- [7] P. Alliez, G. Ucellini, C. Gotsman, M. Attene. *Recent advances in remeshing of surfaces*. Technical report, 2005.
- [8] V. Surazhsky, C. Gotsman. Explicit surface remeshing. *Proc. of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pág. 20-40, 2003.
- [9] S. Dong, S. Kircher, M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer-Aided Geometric Design*, 22, pág. 392-423, 2005.
- [10] F. Massarwi, C. Gotsman, G. Elber. Papercraft models using generalized cylinders. *Proc. of Pacific Conference on Computer Graphics and Applications*, pág. 148-157, 2007.
- [11] I. Shatz, A. Tal, G. Leifman. Paper craft models from meshes. *The Visual Computer*, 22(9), pág. 825-834, 2006.
- [12] J. Mitani, H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *Proc. of ACM/SIGGRAPH*, 23, pág. 259-263, 2004.
- [13] ImageMagick. <<http://www.imagemagick.org/>>.
- [14] M.S. Floater, K. Hormann. Surface parametrization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling*, Ed. Springer, pág. 157-186, 2005.
- [15] F. Tay, T. Chong, F. Lee. Pattern nesting on irregular-shaped stock using genetic algorithms. *Engineering Applications of Artificial Intelligence*, 15, pág. 551-558, 2002.
- [16] S.Q. Xie, G.G. Wang, Y. Liu. Nesting of two-dimensional irregular parts: an integrated approach. *International Journal of Computer Integrated Manufacturing*, 20(8), pág. 741-756, 2007.
- [17] T. Akinen, E. Haines, N. Hoffman. *Real-Time Rendering*. Third Edition. Ed. AK Peters, pág. 537-539, 2008.
- [18] M.M. Oliveira. Correcting texture mapping errors introduced by graphics hardware. *Proc. of the 9th Pacific Conference on Computer Graphics and Applications*, pág. 31-38, 2001.