

## Relleno de huecos en mallas 3D completas mediante algoritmos de restauración de imagen

Emiliano Pérez Hernández<sup>a,\*</sup>, Santiago Salamanca Miño<sup>b</sup>, Carlos Cerrada Somolinos<sup>a</sup>, Pilar Merchán García<sup>b</sup>, Antonio Adán Oliver<sup>c</sup>

<sup>a</sup>Escuela Técnica Superior de Informática. Universidad Nacional de Educación a Distancia. Madrid

<sup>b</sup>Escuela de Ingenierías Industriales. Universidad de Extremadura. Badajoz

<sup>c</sup>Escuela de Informática. Universidad de Castilla la Mancha. Ciudad Real

### Resumen

En este artículo se propone un método para rellenar huecos en objetos 3D. La idea principal es la de utilizar un algoritmo de restauración de imágenes para reconstruir las partes que se han de completar. Para ello es necesario hacer una transformación de los datos de la superficie 3D en datos de imagen 2D. Por tanto, el algoritmo propuesto parte de una etapa inicial de identificación de huecos. A continuación se hace la selección, para cada hueco, de una porción de malla representativa. Seguidamente se calcula el punto de vista o plano de proyección adecuado para obtener una imagen de rango de esa porción de malla. A la imagen de rango obtenida se le aplica el algoritmo de restauración de imagen. Finalmente, se aplica una transformación inversa 2D a 3D y se integra el resultado con la malla inicial. Se trata de un algoritmo robusto, válido para diversos tipos de huecos y diferentes tamaños de los mismos. Copyright © 2012 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

### Palabras Clave:

visión por computador, gráficos por computador, imágenes digitales, imágenes de rango, restauración de imágenes

### 1. Introducción

La creación de modelos tridimensionales es de utilidad para un amplio rango de aplicaciones industriales y es llevada a cabo cada vez más habitualmente mediante el empleo de sensores de rango.

La reconstrucción completa de la superficie 3D a partir de la información adquirida con un sensor de rango puede dividirse en cuatro etapas: la adquisición de vistas parciales, el registro o alineación de esas vistas, la integración de todas las vistas en una única representación y, por último, el procesado de la malla. Esta última etapa es muy importante para conseguir un modelo que presente una corrección geométrica y topológica.

Uno de los tratamientos que se aplica en la etapa de procesado es el de rellenado de huecos. Con este paso se persigue la obtención de una malla hermética, es decir, totalmente cerrada, que resulta de gran utilidad en multitud de aplicaciones: fabricación con sistemas de prototipado rápido, caracte-

rización de superficies, reconocimiento de objetos (Salamanca et al. (2009), Adán and Huber (2011)), etc.

A continuación se realizará una revisión de técnicas de rellenado de huecos que permita contextualizar nuestro trabajo entre los trabajos propuestos por otros autores.

#### 1.1. Revisión de trabajos previos

En la literatura especializada pueden encontrarse una infinidad de técnicas de relleno de huecos en mallas. En Pérez et al. (2010) se propone una revisión de varias de ellas, que enumeramos clasificadas en dos categorías principales:

- Métodos en donde el relleno es un proceso implícito en la creación del modelo 3D.
- Métodos en donde el relleno es un proceso independiente a la creación del modelo 3D.

De los dos tipos de métodos el más empleado es el segundo debido, fundamentalmente, a su versatilidad. El algoritmo de rellenado de huecos propuesto en este trabajo se engloba en esta segunda categoría, por lo que nos centraremos en comentar los trabajos más relevantes dentro de esta categoría.

Entre ellos se pueden destacar el presentado por Davis et al. (2001), que es un método representativo de aquéllos que usan

\* Autor en correspondencia

Correos electrónicos: [emiliano@issi.uned.es](mailto:emiliano@issi.uned.es) (Emiliano Pérez Hernández), [ssalamanca@unex.es](mailto:ssalamanca@unex.es) (Santiago Salamanca Miño), [ccerrada@issi.uned.es](mailto:ccerrada@issi.uned.es) (Carlos Cerrada Somolinos), [pmerchan@unex.es](mailto:pmerchan@unex.es) (Pilar Merchán García), [Antonio.Adan@uclm.es](mailto:Antonio.Adan@uclm.es) (Antonio Adán Oliver)

funciones implícitas para el rellenado de los huecos. La función implícita, de distancia, se define en el entorno del hueco a rellenar y sobre ella se aplica un proceso de difusión que extiende la superficie a lo largo del volumen y cierra el hueco. Éste método presenta muy buenos resultados cuando los huecos son pequeños, independientemente de la curvatura de las superficies que circundan al hueco. Sin embargo, su coste computacional lo hace inaplicable cuando los huecos son grandes.

En Sharf et al. (2004) proponen una técnica basada en los métodos de síntesis de textura que aprovecha la información del contexto, entendido por contexto las zonas con similitud superficial a la del entorno del hueco, para acometer el rellenado de éstos. La ventaja de este método es que en determinados tipos de superficies ofrece una visualización muy realista, aun cuando la superficie inferida en el proceso de rellenado no tiene por qué ser idéntica a la de la pieza original que se haya modelado.

En Wang and Oliveira (2007) se toma una nube de puntos como datos de entrada y generan una representación intermedia consistente en una malla de caras triangulares sobre la que se hace una detección de bordes para determinar los huecos. Tras ello, a partir del grupo de puntos vecinos a esos bordes, se realiza una interpolación mediante la técnica MLS (*Moving Least Squares*) (Lancaster and Salkauskas (1981)). Este método interpola el relleno con datos del entorno, produciendo muy buenos resultados en huecos en zonas relativamente curvadas, aunque presenta la limitación de que el entorno del hueco ha de ser suavizado. Además es un método semiautomático que necesita en algunos momentos la intervención del usuario. De forma similar, en Wu et al. (2008) también se emplean los vértices del entorno del borde de hueco como centros de interpolación para definir una superficie implícita local, que servirá, aplicando las funciones de base radial, para interpolar el contenido del hueco. Los resultados experimentales presentados por los autores han sido obtenidos sobre mallas muy sencillas y en zonas de no gran curvatura, por lo que se puede deducir que los resultados no permiten ser aplicados a mallas y zonas complejas.

El método que se propone en Brunton et al. (2009) realiza un proceso de despliegue de los bordes de los huecos mediante un proceso de minimización de energía. Este hueco desplegado es rellenado mediante una técnica simple de triangulación de Delaunay para, como último paso, volver a ser plegado el borde de hueco y la superficie inferida. Este método presenta varias desventajas. La primera es que no puede ser usada cuando el polígono que define el hueco es complejo. La otra es que al no usar información del entorno del hueco, y usar sólo la información del borde, las superficies que se pueden reconstruir son de curvatura suave.

El proceso de relleno que se propone en Wei et al. (2010) se realiza en tres pasos. El primero es una triangulación del hueco mediante una función que optimiza los ángulos de la triangulación. A continuación, se aplica una subdivisión iterativa para igualar el tamaño de las caras con las del entorno del hueco, finalmente se aplica un filtro laplaciano para suavizar la superficie. Es llamativa la capacidad de resolución de huecos de gran tamaño, aunque en superficies sin grandes complejidades.

En Li et al. (2010) se presenta un método que emplea técnicas de ajuste polinomial, y que tiene como ventaja la preserva-

ción de características de la superficie del entorno del hueco. Además de trabajar bien en zonas angulosas, no excesivamente abruptas, también rellena correctamente huecos en formas organizadas, aunque en huecos no muy grandes y siempre con variaciones suaves de curvatura.

Otro método que destaca en la capacidad para reproducir las características del entorno del hueco es el propuesto en Hu et al. (2012), en el que, en primer lugar, se triangula el hueco mediante un conjunto de nuevos triángulos; a continuación se refina esa porción creada en función de la densidad de vértices del borde de hueco; y finalmente se aplica un filtro bilateral de eliminación de ruido a esa porción para mejorar el resultado y mantener las características de la superficie en el entorno del hueco. Ofrece buenos resultados, aunque en zonas angulosas que presenten un cierto suavizado.

Por último, el procedimiento propuesto en Wang et al. (2012) se realiza a partir de la detección de vértices característicos, que son los pertenecientes a bordes y esquinas. A partir de estos vértices, clasifican los tipos de huecos como simples, que son los que no tienen vértices característicos; y complejos, que son los que sí los tienen. Estos últimos huecos son subdivididos en huecos simples, usando como fronteras entre ellos los vértices característicos. Finalmente, sobre cada hueco simple, se realiza un rellenado plano. Este método permite rellenar huecos en modelos de piezas con cambios bruscos (piezas industriales, por ejemplo), pero no objetos de forma libre en la que es muy difícil que existan estos vértices característicos.

Tabla 1: Comparación entre las características de los distintos métodos referenciados y el método propuesto

Método	Tamaño de los huecos	Relleno en zonas angulosas/concavas	Seguimiento de estructuras
Davis	Huecos no muy grandes.	Muy limitado / Con limitaciones	Con limitaciones
Sharf	Huecos no muy grandes	Sí, si existen zonas similares en la malla / Sí	Sí, si existen zonas similares en la malla
W. and O.	Diferentes tamaños de huecos	No / Solo en huecos simples	Sí, si existen zonas simétricas
Wu	Huecos no muy grandes, según tipo de superficie	No / Solo en huecos simples	No
Brunton	Huecos grandes pero simples	Sí, en huecos simples y formas libres / Idem	No
Wei	Diferentes tamaños, pero simples	No / Limitado a zonas de poca curvatura	No
Li	Huecos no muy grandes	Sí / Siempre que existan estructuras lineales	Sí
Hu	Huecos no muy grandes respecto a estructuras	Sí / Limitado	Sí
Wang	Huecos no muy grandes respecto a estructuras	Siempre que existan estructuras lineales / Idem	Sí
Propuesto	Huecos grandes y pequeños	Sí / Sí, si existe zona de curvatura similar	Sí

Tabla 2: (Continuación de la tabla 1) Comparación entre las características de los distintos métodos referenciados y el método propuesto

Método	Calidad del resultado	Otras consideraciones
Davis	Muy suavizado	Sensible al proceso de adquisición. Coste computacional muy alto
Sharf	Bueno, si existen zonas similares.	Falla con mallas con ruido. Deben existir zonas similares a la de relleno. No siempre captura bien los detalles.
W. and O.	Suavizado	Realiza un remallado completo. El entorno de los huecos debe ser suave. Solo para huecos simples.
Wu	Muy suavizado	Modifica el entorno del borde de hueco. El entorno de los huecos debe ser suave. Solo para huecos simples.
Brunton	Bueno, en zonas con poca curvatura	Válido para huecos simples y superficies con variaciones suaves de curvatura
Wei	Solo apto en zonas con poca curvatura y huecos simples	Solo apto en zonas con poca curvatura y huecos simples. La superficie de relleno puede tener una resolución diferente a la de la malla.
Li	Muy bueno en piezas angulosas. En formas libres, muy suavizado	Usa parámetros fijados de forma arbitraria para cada malla. Produce un ligero suavizado.
Hu	Bueno en piezas de ingenierado. Bueno en huecos simples de formas libres	Solo apto para huecos simples. Produce un ligero suavizado.
Wang	Bueno en piezas de ingeniería.	Solo apto para piezas de ingeniería, en donde existan estructuras lineales suficientemente marcadas.
Propuesto	Bueno, principalmente en objetos de forma libre	El nº de iteraciones no se elige óptimamente. Produce un ligero suavizado en zonas angulosas

En el presente artículo se propone un método de relleno de huecos aplicado a mallas de objetos completos. Este método se encuadraría dentro del segundo tipo de procedimientos anteriormente mencionados y en concreto, dentro de los que utilizan técnicas aplicadas tradicionalmente en fotografías 2D. La idea en la que se basa es en la utilización de técnicas de restauración de imagen 2D (*image inpainting*), para el relleno de huecos en 3D. Por tanto, el algoritmo requerirá un proceso de adaptación entre ambos tipos de datos: 2D y 3D.

Se decidió probar esta idea novedosa, inicialmente sobre vistas parciales y con diversos algoritmos de restauración. Esos trabajos son los que se presentan en Pérez et al. (2008) y Salamanca et al. (2008), donde se parte de una base de datos de vistas parciales, a las que se generan huecos artificialmente. Se rellenan esos huecos aplicando dos métodos de restauración diferentes y se compara el resultado con la vista inicial sin huecos (medida del error, o de las distancias entre malla original y restaurada). El método desarrollado ofrecía buenos resultados en diferentes tipos y tamaños de huecos. Se determinó con qué algoritmo de restauración funcionaba mejor y además se aseveró la bondad del método haciendo una comparación con un software comercial. A partir de ahí, una vez validado con vistas parciales, se decidió diseñar un método de relleno automático de huecos en objetos completos de forma libre, aplicando la misma idea, pero añadiendo las diferentes y complejas

situaciones en que pueden aparecer los huecos en los objetos digitalizados.

En las tablas 1 y 2 se muestra una comparativa de las características de los distintos métodos referenciados en comparación con el método que se expone en este artículo. Sobra decir que, pesar de la diversidad de técnicas, no puede decirse que haya un método de relleno que sea válido para cualquier situación, dado el amplio abanico de éstas: tamaño grande o pequeño de los huecos, huecos planos, huecos con gran curvatura, huecos situados en esquinas, aristas o zonas angulosas, huecos con islas interiores, huecos en zonas de superficie con patrón de repetición, etc. Se trata, por tanto, de plantear un método que ofrezca robustez y versatilidad con respecto a esas posibles situaciones.

El método propuesto se describirá según el siguiente esquema: en la sección 2, se describirá el método de manera general, enumerando las distintas etapas que lo componen. Desde la sección 3 a la 7 se detallarán cada una de esas etapas. Seguidamente, en la sección 8, se presentarán algunos de los resultados experimentales obtenidos y finalmente se expondrán las conclusiones en el apartado 9.

## 2. Descripción general del método

La idea de base del método que se propone, consiste en aplicar un algoritmo de restauración de imágenes, diseñado para datos 2D, a un problema que se presenta en datos 3D, por lo que, como es lógico, habrá que diseñar los mecanismos adecuados para llevar a cabo las diferentes transformaciones entre los datos. El método que se ha desarrollado puede ser resumido en los siguientes pasos:

1. Identificación de los  $h_i$  ( $i = 1, \dots, m$ ) huecos en la malla total  $M_T$ .
2. Selección de la porción de malla  $M_{v_i} \subset M_T$ , que es la parte de malla que se encuentra en el entorno del hueco  $h_i$ .
3. Transformación 3D a 2D para obtener la imagen de rango  $I_{h_i}$ , asociada a la malla  $M_{v_i} \subset M_T$ , correspondiente al entorno de uno de los  $m$  huecos.
4. Aplicación del algoritmo de *image inpainting* sobre  $I_{h_i}$ . La imagen reconstruida se denotará  $I_{f_i}$ .
5. Transformación 2D a 3D de los datos generados por el algoritmo de *image inpainting*. La malla obtenida, que se denota como  $M_{f_i}$ , se fusionará con  $M_T$  para la obtención de la malla final sin huecos  $M'_T$ .
6. Se repite el proceso desde 2, con otro hueco de la lista de los  $m$  huecos, pero ahora utilizando  $M'_T$ .

Antes de comenzar a describir cada una de estas etapas, es necesario recalcar que el método propuesto ha sido desarrollado para ser aplicado en mallas triangulares. A estas mallas se les exigen ciertas características básicas para que el resultado que se obtenga sea óptimo. Éstas son:

- La malla ha de ser una variedad bidimensional: cada arista es compartida como máximo entre dos triángulos.

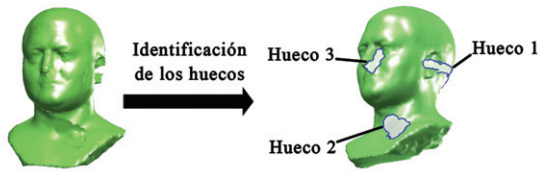


Figura 1: Etapa inicial de identificación de los  $h_i$  ( $i = 1, \dots, m$ ) huecos en la malla total  $M_T$ .

- El tamaño de los triángulos ha de ser aproximadamente constante para toda la malla.

Ambos requisitos no son restrictivos, ya que, a cualquier superficie se le puede aplicar un algoritmo de eliminación de triángulos que no sean una variedad bidimensional y de remallado, antes de iniciar el proceso. Por ello, estas características no restringen en ningún momento su ámbito de aplicación ni restan generalidad al método.

### 3. Identificación de los huecos

La primera etapa del método será, lógicamente, la búsqueda de los huecos que han de ser rellenados y, más concretamente, la determinación de los bordes de esos huecos. Para tal fin, se tiene en cuenta la siguiente definición:

**Definición 1.** Decimos que la malla triangular  $M_T$ , de  $n$  vértices y  $m$  caras triangulares, es una variedad bidimensional cerrada sí y sólo sí, para todas las aristas de la malla formada por dos vértices cada una de ellas,  $\{ \langle v_i, v_j \rangle \mid i, j = 1 \dots n, j = 1 \dots n \wedge i \neq j \}$ , existen dos y solo dos triángulos,  $t_k$  y  $t_l$ , de la malla que comparten dicha arista

A partir de esta definición es directo la definición de un hueco en una malla triangular:

**Definición 2.** Se dice que existe un hueco en una malla triangular  $M_T$ , de  $n$  vértices y  $m$  caras triangulares, sí y solo sí existe un conjunto de aristas que forman parte de un solo triángulo, definida cada una de ellas a partir de un par de vértices  $\langle v_u, v_w \rangle$ , que forman un ciclo o camino cerrado,

$$B_i = \{ \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \dots, \langle v_k, v_{k+1} \rangle, \dots, \langle v_{p-1}, v_p \rangle, \langle v_p, v_1 \rangle \} \quad (1)$$

Por tanto, para buscar las aristas que delimitan los huecos de  $M_T$  habrá que determinar aquellos ciclos  $B_i$  existentes en la malla. A partir de los distintos  $B_i$  se define el conjunto de bordes de huecos de la malla como  $B_m = \cup_i B_i$ . Una vez determinado este conjunto, se procederá al relleno de los huecos, de uno en uno, de forma iterativa, de tal manera que para acometer el relleno del hueco  $i$ , se dispondrá de la malla con  $i - 1$  huecos ya rellenados. En la figura 1 puede observarse un ejemplo de identificación de tres huecos en una malla.

### 4. Selección de la porción de malla

Como se ha comentado en la sección 3, para proceder al relleno es necesario obtener una imagen de rango de la zona en donde aparezca el hueco. Será ella la que se emplee como entrada al algoritmo de reparación de imágenes. Este algoritmo usa la información que circunda al hueco para inferir la información no existente, por lo que la extensión de la zona que se proyectará para generar la imagen tendrá que ser lo suficientemente grande, con respecto al tamaño del hueco, como para que la información que aporte sea suficiente.

Para la determinación de la porción de malla a proyectar, haremos uso del concepto de anillo de una malla triangular.

**Definición 3.** Decimos que  $R_j^i = \{n_1, n_2, \dots, n_k\}$  es el anillo, con  $j$ -ésimo grado de vecindad, de un nodo  $n_i \in M_T$  sí y sólo sí  $\forall n_k \in R_j$ , la mínima distancia, medida como número de aristas, para ir desde  $n_i$  a  $n_k$  es  $j$ .

A partir de esta definición, la determinación de la zona de la malla en torno al hueco es fácil de hacer aplicando la siguiente relación:

$$M_{v_k} = \bigcup_{i=1}^l \bigcup_{j=1}^v R_j^i \quad (2)$$

Donde  $M_{v_k}$  son los nodos en tono al hueco  $B_k$ , con un grado de vecindad  $v$ , calculados a partir de los anillos de los nodos de  $B_k$ . Las aristas de esta porción de malla se corresponde con el subconjunto de aristas de  $M_T$  en donde sus dos vértices pertenezcan a  $M_{v_k}$ . Esta porción de malla cuenta con dos bordes, uno interior que coincide con  $B_k$  y otro exterior  $B_{ev}$ .

El valor que se elija del parámetro  $v$  va a depender de las características de la malla del objeto. Así, si se tiene una malla con baja resolución (triángulos muy grandes), bastará con un grado de vecindad  $v$  pequeño para abarcar una extensión considerable del objeto. Sin embargo, si la resolución es muy elevada se necesitará un valor  $v$  mucho mayor para obtener una porción representativa de la malla  $M_{v_k}$ . Se determinó una expresión empírica, mediante experimentación sobre un conjunto de mallas, de las que se generaron copias de las mismas a diferentes resoluciones. Se buscó encontrar la relación aproximada entre el grado de vecindad, la resolución y el área de porción generada. La expresión que se encontró, y que se probó que era de utilidad en un funcionamiento automático del algoritmo, es la siguiente:

$$v \approx \frac{A_i}{d_m (0,106 \cdot \kappa \cdot A_i + 18)} \quad (3)$$

donde  $A_i$  representa el área de la proyección del hueco  $i$  sobre el plano de proyección,  $\kappa$  es la proporción entre el tamaño del hueco y tamaño de datos válidos en la imagen, que necesita el algoritmo de restauración para producir resultados adecuados ( $\kappa \approx 0,15$ ), y, finalmente,  $d_m$  es la longitud media de las aristas de la malla.

En la figura 2 se representan las diferentes porciones de malla extraídas para cada uno de los huecos mostrados en la figura 1.



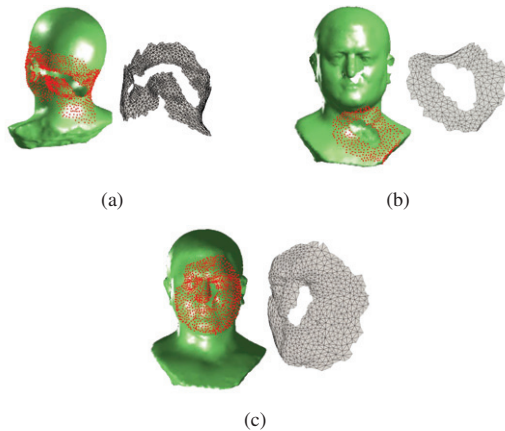


Figura 2: Selección de las porciones de malla para cada uno de los huecos que aparecen en la figura 1: (a) Hueco 1; (b) Hueco 2; (c) Hueco 3. Se han utilizado distintos valores de  $\nu$  en cada caso:  $\nu = 7$ ;  $\nu = 5$ ;  $\nu = 11$ , respectivamente.

## 5. Generación de la imagen de rango

En esta etapa se lleva a cabo la transformación de los datos 3D a 2D, mediante la generación de una imagen de rango,  $I_{h_i}$ , de la malla  $M_{v_i}$ . La imagen de rango es un método de representación de superficies en el que en una imagen se codifica mediante niveles de grises la información de altura de los puntos 3D respecto a un punto de vista o plano de referencia. No es válido cualquier plano o punto de vista para obtener una imagen de rango adecuada de la superficie ya que tanto su posición como su orientación con respecto a la superficie 3D son decisivas para que la proyección resultante sea representativa de una porción de malla. Por otra parte, el otro parámetro decisivo para la generación de la imagen de rango es el ancho de cada una de las celdas (o píxeles de la imagen) en que se divide el plano de referencia. Por tanto, a continuación van a explicarse esos dos parámetros críticos en el proceso de creación de la imagen de rango:

1. Plano de proyección.
2. Resolución de la imagen de rango.

### 5.1. Plano de proyección

En este punto lo que se pretende es elegir qué orientación y tamaño deberá tener el plano de referencia con respecto a la superficie 3D. Como se ha mencionado, el plano que se elija tendrá influencia en la capacidad de representación que tenga la imagen con respecto a los datos 3D.

Dado que la región de interés es el propio hueco que ha de ser rellenado, ha de elegirse una orientación según la cual el hueco sea “visto” de la mejor manera posible. Para ello, se utilizará el criterio de área máxima. Según este criterio, la mejor orientación es aquella en la que la proyección del borde del hueco  $B_k$  da lugar a un polígono en el plano,  $P_k$ , cuya área sea máxima (figura 3).

Para resolver este problema es necesario definir un nuevo sistema de referencia  $\Sigma'(X'Y'Z')$  ligado al plano de referencia de la imagen de rango, donde los ejes  $X'$  e  $Y'$  están contenidos en el mismo, es decir, serán los que definan el plano sobre el que

se proyectará; y el eje  $Z'$  es perpendicular a ambos, apuntando en dirección hacia el objeto. En el caso de que la proyección del borde del hueco fuese un polígono simple, es decir diese lugar a un polígono en donde no existiesen cruces entre sus lados, sería suficiente con que el plano  $\{X', Y'\}$  fuese aquel que mejor se aproximase a los nodos que definen el borde del hueco, ya que es éste el que mayor área proporcionaría. Sin embargo, las proyecciones suelen dar lugar a polígonos complejos (polígonos en los que existen cruces entre algunos de sus lados) (figura 4), y en este caso el plano de proyección no tiene por qué coincidir con el plano que se comentaba anteriormente. Es por ello que la resolución de este problema se plantea como un problema de maximización, que buscará que  $\Sigma'$  tenga una orientación tal, que el área del polígono  $P_k \subset X'Y'$  sea máxima (figura 3).

La definición de la funcional a maximizar es la siguiente:

$$\text{máx}[A_{P_k}(\Sigma')] \quad (4)$$

sujeto a:

$$\begin{aligned} |\widehat{x'}| &= 1 \\ |\widehat{y'}| &= 1 \\ (\widehat{x'} \perp \widehat{y'}) &\perp \widehat{z'} \end{aligned}$$

donde  $A_{P_k}(\Sigma')$  es el área del polígono  $P_k$  contenido en el plano  $X'Y'$  del sistema de referencia  $\Sigma'$ ; y  $\widehat{x'}$ ,  $\widehat{y'}$ ,  $\widehat{z'}$  son los vectores unitarios de los ejes  $X'$ ,  $Y'$  y  $Z'$ , respectivamente. Para la resolución del problema, se emplea el algoritmo de Levenberg-Marquardt.

Como es bien conocido, la convergencia del algoritmo depende fundamentalmente de que el valor inicial del que se parte en la resolución del problema sea los más cercano a la solución final. Como se ha comentado en el párrafo anterior, en el caso de que el polígono fuese simple, el plano que proporciona la mejor solución es el plano que mejor se aproxima a los nodos del borde. En el caso de que el polígono sea complejo, no será éste, pero sí podemos esperar que sea cercano a dicho plano, por lo que el valor inicial, o plano de proyección inicial, que se empleará será el plano que mejor se aproxima a los nodos del borde del hueco.

Una vez obtenido el  $\Sigma'$ , se aplica una transformación  $T$  a  $M_{v_i}$  para obtener sus coordenadas con respecto a dicho sistema de referencia.

Con respecto al tamaño del plano de referencia, o lo que es lo mismo, el tamaño de la imagen de rango, viene determinado por el tamaño de la porción de malla que se ha de proyectar.

A partir de esta nueva representación en  $\Sigma'$  se determina el tamaño de la imagen de rango. Así, si la porción de malla  $M_{v_i}$  está formada por  $C$  coordenadas, cuyos valores máximos y mínimos en  $X'$  y en  $Y'$  son  $x_{\text{máx}}$ ,  $y_{\text{máx}}$  y  $x_{\text{mín}}$ ,  $y_{\text{mín}}$ , respectivamente, el tamaño que representa la imagen es  $w \times h$ , en donde:

$$\begin{aligned} w &= x_{\text{máx}} - x_{\text{mín}} \\ h &= y_{\text{máx}} - y_{\text{mín}} \end{aligned} \quad (5)$$

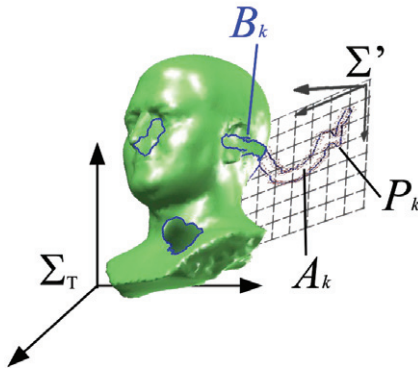


Figura 3: Determinación de la orientación de  $S'$  ligado al plano de referencia que haga que el área de  $P_k, A_k$ , sea máxima.

### 5.2. Resolución de la imagen de rango

Una vez que se ha determinado la orientación y el tamaño del plano de proyección, el siguiente paso es la definición de la resolución de la imagen  $I_{h_i}$  en píxeles.

Las expresiones representadas en (5) definen un área rectangular que ha de dividirse en una cuadrícula o rejilla, cuyas celdas se corresponden con los píxeles de la imagen  $I_{h_i}$ .

El número de celdas de la rejilla de proyección determina la resolución y la discretización de la proyección  $M_{v_i}$ . De esta manera, si se elige que la anchura de la rejilla es  $q$  ( $mm/pixels$ ), la resolución de la imagen es  $1/q$  ( $pixels/mm$ ) y su tamaño es  $(n, m) = (w/q \times h/q)(pixels)$ . El píxel  $(f, k)$  almacena la información de profundidad en el nuevo  $\Sigma'$ , o lo que es lo mismo, la coordenada  $z_i$  del nodo  $n_i \in M_{v_i}$  cuya coordenada  $(x_i, y_i)$  verifica:

$$\begin{aligned} (k-1) \cdot q &\leq x_i < k \cdot q \\ (f-1) \cdot q &\leq y_i < f \cdot q \end{aligned} \quad (6)$$

En Pérez et al. (2008) y Salamanca et al. (2008) se expone la metodología que se desarrolló para la determinación de la resolución óptima de  $I_{h_i}$ . En esas publicaciones se muestra que el valor óptimo es un valor que está relacionado proporcionalmente con la media de las longitudes de las aristas de  $M_{v_i}$ . Para su determinación se realizó una experimentación sobre una base de datos de vistas parciales, de las que se obtuvieron diversas imágenes de rango para diferentes resoluciones.

### 5.3. Traslación y escalado de la imagen de rango

Un parámetro que aún no ha sido tratado y que también ha de tenerse en cuenta es la posición relativa de la vista parcial con respecto al plano de proyección, en concreto, en lo referente a la magnitud de la distancia entre ambos. Aunque la imagen de rango puede generarse igualmente para cualquier valor de esa distancia, el algoritmo de restauración sí es sensible a ese valor, o lo que es lo mismo, al rango de valores que almacene la imagen que tiene que procesar. De esta forma, si la distancia es

alta, los valores almacenados en la imagen de rango son elevados y se necesitarán más iteraciones para conseguir un valor final adecuado en las zonas de relleno. Por ello interesará acercar lo máximo posible la vista parcial hacia el plano de referencia. Esto se conseguirá realizando dos operaciones consecutivas a la malla: una traslación y un escalado. Para ello, si partimos de una malla cuyos vértices tienen las coordenadas  $(x_i, y_i, z_i)$  en el sistema de referencia  $\Sigma_i$ , se sabe que la distancia a la que se encuentra la malla del plano de referencia es:

$$d_i = \min(z_i) \quad (7)$$

Por tanto, la primera operación que se realizará será una traslación en  $Z$  de  $-d_i$  unidades. De esta forma, las nuevas coordenadas de la malla serán:

$$(x'_i, y'_i, z'_i) = (x_i, y_i, z_i - d_i) \quad (8)$$

Por último quedará aplicar un factor de escala en  $Z$ ,  $f_{esc}$ , para que el rango de valores en  $Z$  disminuya. Con lo cual, las coordenadas finales de la malla serán ahora:

$$(x''_i, y''_i, z''_i) = (x_i, y_i, z'_i \cdot f_{esc}) \quad (9)$$

Todas las operaciones han sido explicadas haciendo referencia a la representación tridimensional de la porción de malla, es decir, mediante sus coordenadas  $x, y, z$ . Pero, como se ha mencionado, esas operaciones sólo se han aplicado en la coordenada  $Z$  de los nodos de la malla, que es la que es almacenada y representada por la imagen de rango de la malla. Por tanto, bastará con aplicar las operaciones sobre la imagen de rango que se ha calculado,  $I_h$ . De tal forma que la imagen de rango final,  $I'_h$ , será:

$$I'_h = (I_h - \min(I_h)) \cdot f_{esc} \quad (10)$$

Lógicamente, estas dos operaciones habrán de ser tenidas en cuenta, tras la aplicación del algoritmo de *image inpainting*, en el paso de transformación 2D a 3D que se indicaba en la sección 2.

### 5.4. Interpolación de la imagen de rango

Dado que, generalmente, no se tendrán mallas con un tamaño de triángulos totalmente uniforme, será inevitable que en la imagen se produzcan pérdidas. Para reducir su efecto en la información que la imagen de rango ofrece al algoritmo de restauración, se aplica un proceso de interpolación a la imagen de rango obtenida,  $I'_h$ .

Esta interpolación de pequeñas zonas sin datos se hará relleno, mediante un mecanismo de difusión hacia el interior, los valores promedio de los píxeles que se encuentran en la vecindad del hueco. Esta interpolación se hará iterativamente, de

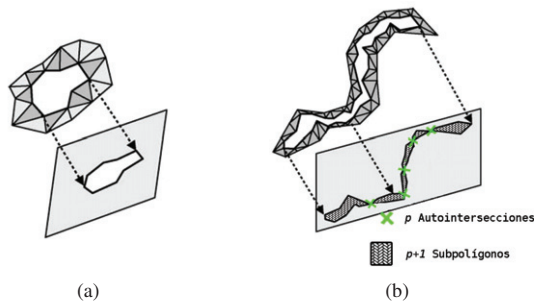


Figura 4: Representación de dos tipos de huecos, en función de la proyección de su borde sobre el plano de referencia: (a) Hueco simple: su proyección no tiene autointersecciones; (b) Hueco complejo: su proyección da lugar a un polígono con autointersecciones.

forma que una vez rellena una zona, esta nueva imagen generada será utilizada en el relleno de la siguiente zona. El criterio de prioridad para acometer la interpolación del conjunto de zonas será el número de píxeles que las componen. Así, se comenzará relleno de las zonas con menor número de píxeles para acabar con las de mayor número.

### 5.5. Huecos simples y huecos complejos

En el problema que se está estudiando es necesario establecer una diferenciación entre dos posibles circunstancias que pueden producirse por el hecho de llevar a cabo la proyección del borde del hueco sobre el plano de referencia. Este hecho ya se comentó en la sección 5.1. La primera situación es que esa proyección dé lugar a que el borde de hueco se convierta en un polígono en el plano sin ninguna autointersección. En ese caso, se dirá que el borde de hueco es simple. La segunda situación que se puede presentar es que la proyección del borde de hueco genere un polígono con autointersecciones, ante lo cual se dirá que el borde es complejo. En la figura 4 aparecen sendas representaciones de las dos situaciones enumeradas.

La aparición de un borde simple o complejo puede condicionar la continuación del proceso de relleno. Concretamente, si el hueco es complejo habrá que modificar el proceso de relleno normal. Esta modificación va a consistir principalmente en la división del relleno del hueco complejo en varios subprocesos de relleno, según el número de autointersecciones que se presenten. La forma de proceder puede ser esquematizada en las siguientes etapas:

- Se tiene un borde de hueco  $B_H$  que se proyecta sobre el plano de referencia elegido, dando lugar al polígono  $P_H$ . En la figura 5 se representa un ejemplo de una malla que presenta un hueco alargado, que es visible desde las dos vistas laterales, derecha e izquierda y la vista trasera. Esto hace intuir que es probable que por su proyección sobre un plano, será un hueco complejo.
- Se comprueba si los lados del polígono intersectan entre sí, es decir, si  $P_H$  tiene autointersecciones. Si no las tiene, el proceso de relleno continúa normalmente. Si, por el contrario, se detectan  $p$  intersecciones, podrá dividirse  $P_H$  en  $p + 1$  subpolígonos,  $P_{S_i}$ , cuyos puntos de unión

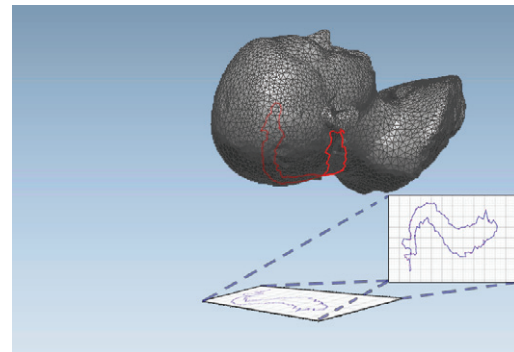


Figura 5: Hueco alargado con un borde que resulta ser complejo con su correspondiente proyección.

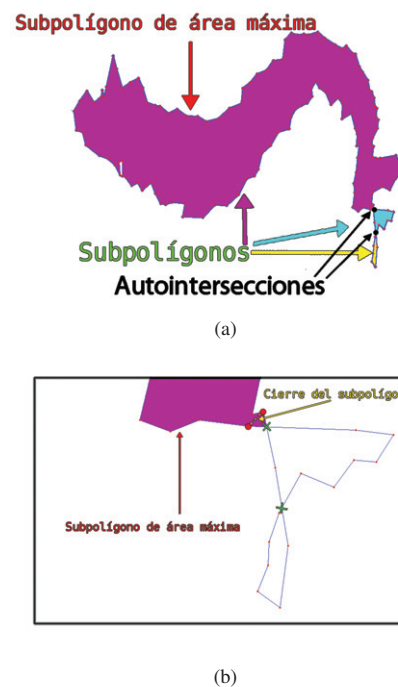


Figura 6: (a) Polígono que se obtiene tras la proyección del hueco representado en la figura 1. Se determinan las dos autointersecciones; Los puntos de intersección dividen el polígono en tres subpolígonos. Se calculan las áreas y se determina cuál es la máxima; (b) El nuevo borde de será el subpolígono con área máxima, aunque para el cierre se unen los dos vértices del subpolígono anteriores al punto de intersección.

son los  $p$  puntos de intersección. En la figura 6(a) se comprueba que el polígono resultante de la proyección, tiene dos autointersecciones.

- Se calculan las áreas de los  $p + 1$  subpolígonos y se busca el polígono con el área mayor,  $P_{Sk}$ . Para el hueco del ejemplo se comprueba que existe uno de los subpolígonos con un área considerablemente mayor que la de los otros dos subpolígonos, como se muestra en la figura 6(a).
- Se toman todos los vértices de  $P_{Sk}$  con excepción del o los puntos de intersección. Ahora se tendrá un polígono abierto por una o dos partes, que se cerrará a continuación directamente uniendo los extremos abiertos. En la figura 6(b) se representa una ampliación de la zona donde aparecen las autointersecciones. Puede observarse en tal figura que el nuevo borde de hueco se cierra utilizando los puntos del subpolígono mayor, que se localizan antes del punto de intersección.
- Se toman los vértices de  $B_H$  correspondientes al polígono que se acaba de construir, y se genera el nuevo borde  $B_{H1}$  con el que continuar el proceso de relleno. Antes de continuar habrá que recalcular la orientación del plano de orientación con el criterio que se estime. El nuevo borde de hueco no contendrá autointersecciones.

Como se deduce inmediatamente, esta operación va a conseguir que al final del proceso sólo se rellene parte del hueco correspondiente a  $B_H$  (ver el resultado para el ejemplo en las figuras 7(a), 7(b) y 7(c)). Por consiguiente, a continuación se iniciará el proceso de relleno, en el que habrá un hueco que no es sino la parte que queda por rellenar. Para ese hueco se volverá a calcular la orientación del plano de proyección, con uno de los criterios anteriores. Si tras fijar dicho plano, el hueco sigue siendo complejo, se procederá de igual forma que en el caso anterior, subdividiendo el hueco. En caso contrario se continúa con el proceso. Se observa en la figura 7(d) el polígono resultante tras la proyección del nuevo borde de hueco generado, tras la partición del hueco. El resultado final del relleno del hueco puede ver en la figura 7(e).

## 6. Aplicación de los algoritmos de restauración

En esta etapa se va a aplicar el algoritmo de *image inpainting* sobre la imagen  $I_{h_i}$  obtenida en la etapa anterior. El empleado es algoritmos de Roth y Black (Roth and Black (2005)). Este algoritmo aplica la idea de codificación de imagen para el aprendizaje de los parámetros de los campos de Markov (Markov Random field, MRF). El método utilizado se denomina Campo de Expertos (*Field of Experts, FoE*) y modela la probabilidad de una imagen en términos de un campo aleatorio con parches sobrepuestos cuyos potenciales son representados como un Producto de Expertos (*Product of Experts, PoE*) Hinton (1999).

La objetivo del Producto de Expertos es modelar la probabilidad de distribución de dimensión elevada (parches de imágenes) mediante el producto de varias distribuciones de expertos,

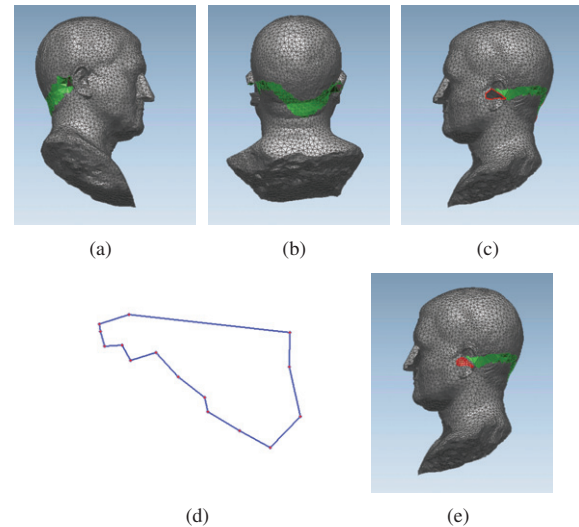


Figura 7: (a), (b) y (c) Diferentes vistas del relleno parcial del hueco de la figura 1, tras la obtención del nuevo borde de hueco de la figura 7(d); (d) Proyección del borde del nuevo hueco generado; (e) Relleno del nuevo borde de hueco, que supone el relleno del hueco completo inicial.

en donde cada experto trabaja en un subespacio de baja dimensión, más fácil de modelar. Concretamente, al aplicar un filtro lineal sobre un parche de imagen se produce un subespacio de una dimensión que hay que modelar. Por tanto, la densidad de probabilidad de un parche de imagen puede ser definida como el producto de las distribuciones que producen los filtros lineales.

En general este tipo de algoritmos no hace una selección automática de los píxeles sobre las que se efectuará el relleno. Sólomente actuarán sobre los que les vengán marcados a la entrada. Por tanto, la aplicación de estos algoritmos sobre la imagen de rango,  $I_{h_i}$ , viene limitada por una máscara de selección de los píxeles a rellenar,  $Q_{h_i}$ . Esta máscara es una imagen binaria, del mismo tamaño que la imagen de rango,  $(n, m)$  píxeles, y en las que aparecen activos los píxeles sobre los que se pretenda realizar el relleno. se muestra en la figura 8.

Un parámetro importante del algoritmo de restauración es el número de iteraciones, que influye tanto en la bondad de la restauración como en el tiempo de cálculo global del algoritmo. Para la elección de este número tienen influencia diversos elementos, difícilmente caracterizables: tamaño de las zonas a restaurar, complejidad de la imagen, situación del hueco en la imagen... Este parámetro limita, por tanto, el diseño de un algoritmo óptimo totalmente automático. De cualquier forma, se ha comprobado que valores ligeramente por encima al millar, sin significar, un incremento excesivo del tiempo de cómputo, generan muy buenos resultados. Por tanto, se fijará un valor del parámetro de ese orden, y sólo en caso, de superficies muy variables, el usuario modificará ese valor.

Por tanto, una vez aplicado el algoritmo de *image inpainting* se dispone de una imagen de rango restaurada, en la que a las zonas correspondientes a los huecos, inicialmente vacías, se les ha sido asignado un valor.



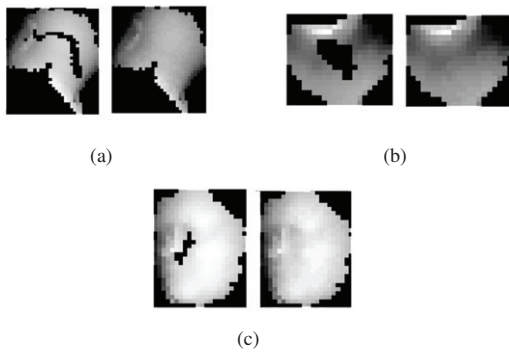


Figura 8: Representación de las imágenes de rango extraídas de las porciones de malla  $M(v_i)$  (a la izquierda) y de las imágenes de rango obtenidas tras la aplicación del algoritmo de restauración, para los huecos 1, 2 y 3 en (a), (b) y (c), respectivamente.

## 7. Integración de los resultados

Para completar la reparación de la malla, es necesario, transformar los datos que se han obtenido en la etapa anterior a 3D. Solamente habrá que aplicar esa transformación a los datos nuevos que se han generado tras la ejecución del algoritmo de restauración, es decir, sobre los huecos rellenos. La selección de estos píxeles en la imagen restaurada es inmediata ya que se obtienen directamente al aplicarle la máscara de selección.

La transformación de 2D a 3D se traduce en la obtención de la coordenada 3D que se corresponde con esos píxeles rellenos. En el proceso de creación de la imagen de rango se decía que el valor que almacenaba cada píxel se correspondía a la distancia a que se encontraba el punto 3D cuya proyección ortogonal sobre la rejilla de proyección incidía sobre esa celda o píxel. Por tanto, el valor que almacene cada píxel relleno fija la distancia a que se encuentra el punto 3D del plano imagen, y, en la representación respecto a  $\Sigma'$  se corresponde con la coordenada  $z'$ . Las coordenadas  $x'$  e  $y'$  vendrán definidas por la posición que tenga ese píxel dentro del plano imagen. De esta forma, si la imagen de rango representaba valores entre una  $(x_{min}, y_{min})$  y una  $(x_{max}, y_{max})$ , y se eligió una resolución de  $q$  píxeles por milímetro, dado un píxel  $(f, k)$  que almacena el valor  $r$ , la correspondiente coordenada 3D para ese píxel será:

$$(x', y', z') = (x_{min} + k \cdot q, y_{max} - f \cdot q, r) \quad (11)$$

Hay que señalar que en la ecuación anterior se ha considerado que las coordenadas de los píxeles vienen definidas por la fila  $f$  y columna  $k$  que ocupan en la matriz imagen. Es por ello que la coordenada  $x$  depende de la segunda coordenada del plano imagen,  $k$ , y la coordenada  $y$  de la primera,  $f$ .

Una vez calculadas todas las coordenadas se dispone de una nube de puntos 3D, correspondientes al hueco que ha sido relleno. Esa nube de puntos está expresada en el sistema de coordenadas  $\Sigma'$ , solidario al plano de referencia, por lo que hay que aplicarle la transformación inversa  $T^{-1}$  para obtener su representación en el sistema de referencia inicial  $\Sigma_i$ . Además habrá que realizar las operaciones inversas a la traslación y escalado que se aplicaron en la ecuación (10).

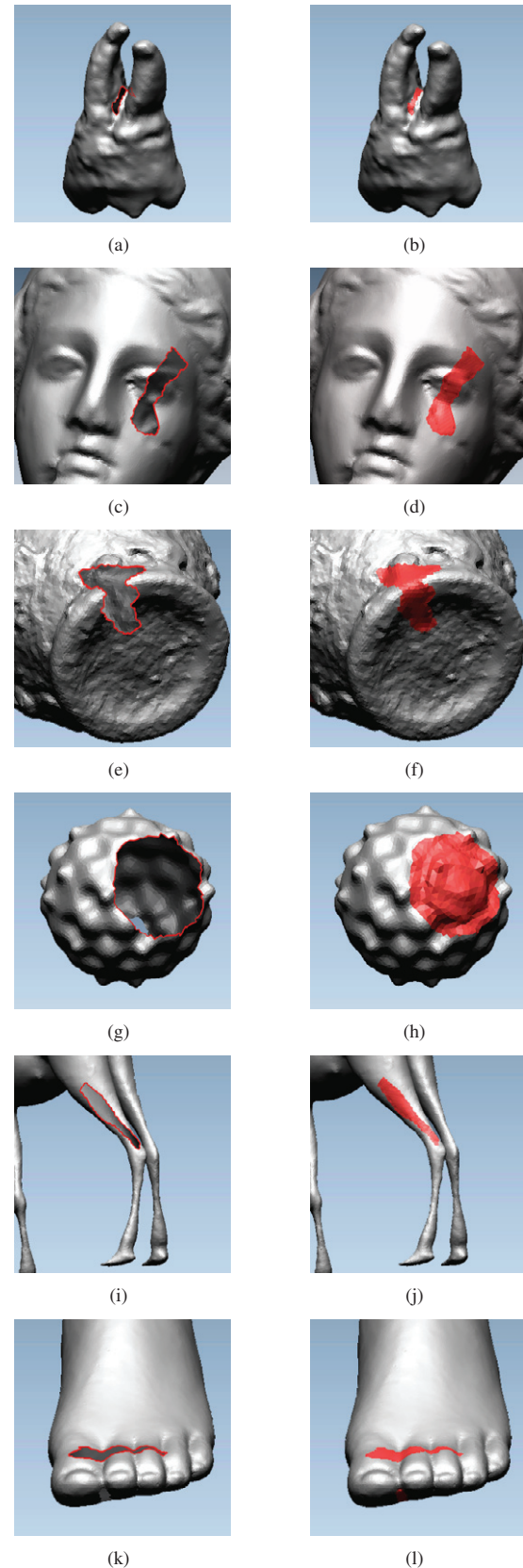


Figura 9: Algunos resultados experimentales obtenidos para el relleno de huecos en objetos completos.

Ahora es el momento de definir las superficies rellenadas mediante la triangulación de esas nubes de puntos. A estos nodos se le añadirán, para calcular la triangulación, los puntos que conforman el borde de los huecos,  $B_h$ . Aunque estos puntos pertenecen a la malla inicial,  $M_T$ , sirven como delimitadores de cada una de las zonas rellenadas. A su vez, tendrán la función de servir como eje de unión o cosido entre la malla inicial con huecos,  $M_T$ , y las mallas de relleno,  $H_c$ .

## 8. Resultados experimentales

El método que se propone fue programado en Matlab y para comprobar su funcionamiento se ejecutó sobre un equipo PC Intel Dual Core 3,4 GHz con 2 GB de RAM, aplicándolo sobre un conjunto de mallas del repositorio AIM@SHAPE Shape Repository AIM@SHAPE (2007). En este conjunto había mallas completamente cerradas, a las que se les generaron huecos de forma artificial y mallas a las que no se le había aplicado un procesamiento final tras su adquisición y, por tanto, presentaban huecos. Estos huecos, sobre los que se aplicó el método, eran de diversa forma y tamaño, y se presentaban sobre superficies de topología variada.

Como se mencionó en la sección 1, inicialmente se probó el funcionamiento del método con una base de datos de vistas parciales, cuyos resultados pueden verse en Pérez et al. (2008) y Salamanca et al. (2008). En estos trabajos se hizo una comparativa del error que ofrecía el método con respecto al relleno efectuado mediante un software comercial. El resultado fue que el método diseñado generaba un error menor en el 75 % de las mallas de la base de datos. Además, con respecto al área de los huecos, a medida que ésta era mayor, nuestro método presentaba un error mucho menor que el software.

De los resultados obtenidos con objetos completos, es destacable, por ejemplo, el relleno del tipo de huecos como el que se muestra en las figuras 9(a) y 9(b), situado en una zona interior o hendidura de la malla, topología habitual en la que aparecen huecos usualmente en la digitalización de mallas. Tras un primer análisis del método de relleno, podría pensarse que, por basarse en una proyección del entorno del hueco, una de las limitaciones que presenta es el relleno de huecos cuyo entorno presenta una gran curvatura, tales como los que aparecen en el interior de hendiduras. Sin embargo en el ejemplo mencionado se demuestra que en tales situaciones el proceso genera una malla de relleno satisfactoriamente. Un caso parecido, aunque a la inversa, es el hueco que aparece en las figuras 9(i) y 9(j), el cual está sobre una superficie casi cilíndrica y en la cual se podría pensar que no es posible tampoco aplicar el método.

Otra característica a resaltar es la capacidad para mantener las estructuras lineales en el entorno donde aparece el hueco, tal y como ocurre en el hueco mostrado con la ceja de la malla de las figuras 9(c) y 9(d). Otro ejemplo similar es el de las figuras 9(k) y 9(l). Podría mejorarse el método, en ese sentido, añadiéndole la capacidad de detección de patrones repetidos y de copia en el interior del relleno de porciones de la malla similares, como la propuesta de Sharf et al. (2004), para, de esta forma, mejorar el resultado en huecos como el mostrado en las figuras 9(g) y 9(h).

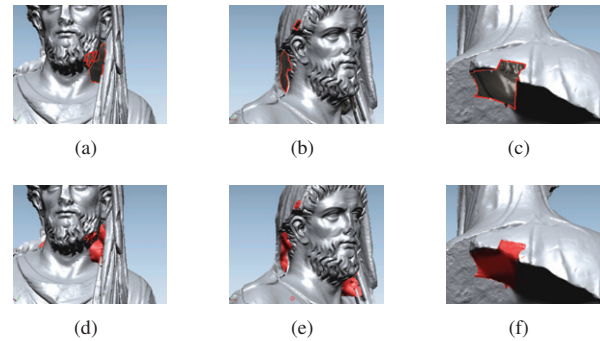


Figura 10: Relleno de la malla de una de las esculturas adquiridas en el Museo Nacional de Arte Romano de Mérida.

Otra situación en la que no todos los métodos de relleno generan soluciones satisfactorias, es en el relleno de huecos situados en aristas de una superficie, con cambio brusco de curvatura. En las figuras 9(e) y 9(f), se observa que el resultado generado con el método propuesto da lugar a que se mantenga la continuidad en la arista o borde. Hay que reconocer aquí que el método de relleno siempre genera mallas con un cierto suavizado, por lo que no podría reproducir una arista excesivamente abrupta.

Además, también se ha aplicado el método sobre diversas mallas obtenidas tras la digitalización de esculturas en el Museo Nacional de Arte Romano de Mérida, en las que aparecieron huecos, debido a algunas limitaciones o errores en el proceso de adquisición. Como ejemplo de este tipo, en la figura 10 se muestran los resultados obtenidos para una de las esculturas.

Dado que en las mallas que presentaban huecos reales, no se tenía una malla de referencia sin errores con la que comparar la bondad de los resultados, se calculó y comparó la variación de la curvatura normal y gaussiana que presentaban, antes y después de aplicar el método de relleno. En la figura 11 se muestran las curvaturas calculadas para el objeto que se muestra en la figura 1. Como puede verse de forma intuitiva, la corrección del relleno mediante este tipo de representación se pone de manifiesto cuando existe una continuidad en las zonas con colores similares, o lo que es lo mismo, con curvaturas aproximadamente iguales. En el caso de objetos completos con huecos reales, producidos durante la fase de adquisición de los mismos, los huecos son, normalmente, de tamaño pequeño en relación al del objeto completo. Se presentan principalmente en hendiduras que son de difícil acceso para su adquisición. En la figura 12 se muestra un ejemplo de relleno de una malla con huecos reales.

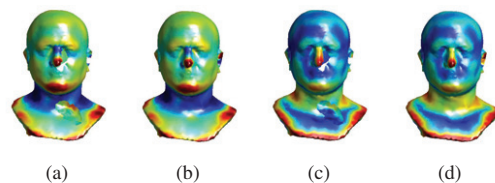


Figura 11: Comparación entre las curvaturas gaussianas y total, antes, en (a) y (c), y después, en (b) y (d), de aplicar el método de relleno, respectivamente.

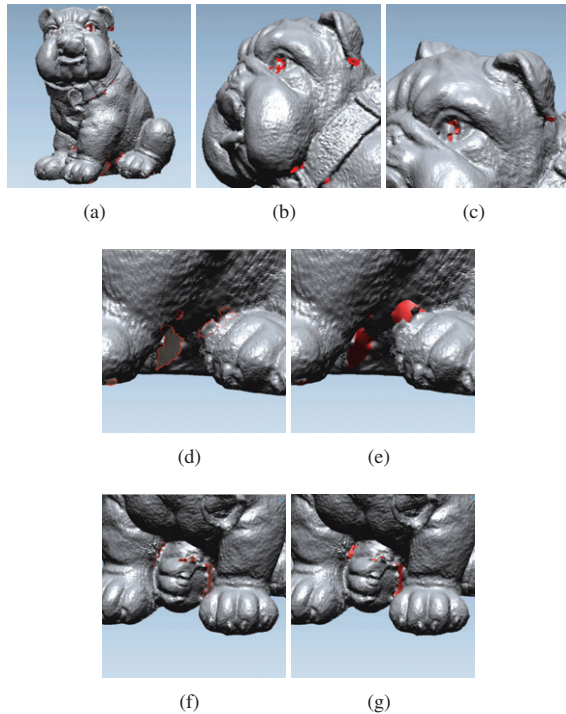


Figura 12: Relleno de la malla *bulldog*. Los huecos que presenta son debidos a limitaciones en el proceso de adquisición, por lo que se sitúan principalmente en hendiduras de la malla.

También se recopiló, para cada una de las mallas, información del proceso de relleno como son: longitud media de las aristas de la malla, áreas de los huecos rellenados, tiempos de cada etapa del proceso y tiempo total de relleno. En las tablas 3, 4 y 5 pueden observarse los datos obtenidos para la malla mostrada en la figura 12.

Hay que señalar que en la fase de diseño del método de relleno propuesto no se buscó una optimización del tiempo de ejecución. De hecho, como se ha mencionado, los algoritmos han sido programados en MATLAB, que, como es bien sabido, no es la opción que consigue precisamente unos tiempos de ejecución óptimos. Aun así, como puede observarse en las tablas anteriormente referenciadas, los tiempos totales de relleno, son del mismo orden que otros métodos de relleno. Incluso son similares a los tiempos de relleno obtenidos mediante el software comercial que se proponía en Pérez et al. (2008). Tras un breve análisis de los datos tabulados, se observa que uno de los parámetros que influye, de manera importante, en el tiempo de relleno es el proceso de inpainting y, en concreto, el número de iteraciones que se seleccionen para el algoritmo de restauración. En esta fase de diseño de nuestro método, ese es un parámetro que no ha sido optimizado.

## 9. Conclusiones

En este artículo se ha presentado un método original para resolver el problema del relleno de huecos, que tiene como base de funcionamiento el uso de técnicas de restauración de

Tabla 3: Detalles generales de los parámetros aplicados para el relleno de la malla de la figura 12. Unidades en mm.  $d_m$  es la longitud media de las aristas.

$d_m$	Área	Nº Iteraciones	Nº Huecos
0,614	82755,214	1000	15

Tabla 4: Detalles de algunos de los resultados obtenidos para la malla de la figura 12. Unidades en mm y seg.  $T_p$ ,  $T_{im}$ ,  $T_{inp}$  y  $T_T$  son los tiempos: de generación de la porción de malla, de la generación de la imagen de rango, de aplicación del inpainting y de relleno total de la malla, respectivamente;  $A(p)$  y  $A(h)$  son las áreas de la porción de malla utilizada y del hueco, respectivamente y  $d_m$  la longitud media de las aristas.

Hueco	1	2	3	4	...
$d_m$	0,528	0,529	0,548	0,531	...
$A(p)$	701,75	541,125	850,625	528,625	...
$A(h)$	45,375	10,375	38,375	5,25	...
$T_p$	0,054	0,04	0,067	0,033	...
$T_{im}$	0,238	0,163	0,376	0,176	...
$T_{inp}$	0,068	0,029	0,055	0,025	...
$T_T$	0,359	0,232	0,499	0,235	...

Tabla 5: Resumen de los tiempos totales de duración de cada uno de los subprocesos del proceso de relleno de la malla de la figura 12. Unidades en seg.  $T_p$ ,  $T_{im}$ ,  $T_{inp}$  y  $T_T$  son los tiempos: de generación de la porción de malla, de la generación de la imagen de rango, de aplicación del inpainting y de relleno total de la malla, respectivamente.

$T_p(M_T)$	$T_{im}(M_T)$	$T_{inp}(M_T)$	$T_T(M_T)$
1,026	3,986	1,739	6,751

imágenes, tradicionalmente aplicadas en fotografías. Se ha demostrado que muchas de estas técnicas resuelven de forma muy efectiva el problema de restauración, o, lo que es lo mismo, de regeneración de información que se ha perdido o deteriorado en una imagen. Por tanto, se ha decidido adaptar este tipo de técnicas para la recuperación de información o relleno de huecos en objetos 3D.

Dado que se plantea utilizar técnicas que, usualmente se aplican a datos bidimensionales, con los datos de superficie 3D de objetos, es necesario estudiar cómo realizar la conversión de un tipo de datos al otro. En ese sentido, se ha desarrollado un procedimiento para la conversión óptima de los datos 3D a un formato de imagen sobre el que se pueda aplicar el algoritmo de restauración.

Ya se demostró en Pérez et al. (2008) la robustez que ofrecía, generando buenos resultados para huecos de diversos tamaños y en una gran variedad de configuraciones en vistas parciales. Los resultados para objetos completos confirman de nuevo la robustez del método de relleno.

Cabe destacar también la capacidad del método para el relleno de huecos en zonas con esquinas o cambios de curvatura brusco, aunque, sí es cierto, que, en esos casos, podría mejo-



rarse la respuesta ante esquinas muy marcadas, puesto que el método propuesto tiende a generar una superficie con variaciones más suaves

En cuanto al seguimiento de estructuras lineales que aparezcan en la superficie, el algoritmo responde correctamente.

Se ha planteado como trabajos futuros, la adaptación del método para una aplicación concreta: el relleno de huecos en nubes de puntos procedentes del escaneado de interiores de edificios.

Además, se plantea también el estudio más detallado de la relación entre el número de iteraciones que necesita el algoritmo de restauración y las características de la malla a rellenar, así como el tamaño del hueco y la porción de malla.

### English Summary

#### Filling holes in digitized 3D meshes using image restoration algorithms

#### Abstract

In this work a method for filling holes in 3D meshes based on a 2D image restoration algorithm is expounded. To do that, data must be converted to a suitable input format, a 3D to 2D transformation is executed by projecting the 3D surface onto a grid. Therefore, the proposed algorithm starts by a first stage of holes identification. Then, a meaningful mesh portion is chosen for each hole. Afterward, the suitable plane of projection must be computed to get the range image of the mesh portion. Later, it is applied the restoration image algorithm to the range image. Finally, an inverse transformation 2D to 3D is performed and the new produced data are integrated with the initial mesh. The result is a robust algorithm which works correctly with several kind of holes and for different sizes of them.

#### Keywords:

computer vision, computer graphics, digital images, range images, images restoration

#### Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos del Ministerio de Ciencia e Innovación DPI2008-05444, DPI2009-14024-C02-01, de la Junta de Extremadura PRI09C088 y de la CAM S2009/DPI-1559 (Robocity2030-II).

### Referencias

- Adán, A., Huber, D., 2011. Análisis de datos 3d para generación automática de modelos bim de interiores habitados. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 8 (4), 357–370.
- AIM@SHAPE, Marzo 2007. Aim@shape shape repository. <http://shapes.aimatshape.net/>.
- Brunton, A., Wuhler, S., Shu, C., Bose, P., Demaine, E., 2009. Filling holes in triangular meshes by curve unfolding. In: *Proceedings of the 2009 IEEE International Conference on Shape Modeling and Applications (SMI'09)*. pp. pp. 66–72.
- Davis, J., Marschner, S., Garr, M., Levoy, M., 2001. Filling holes in complex surfaces using volumetric diffusion. In: *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization and Transmission*. pp. pp. 428–438.
- Hinton, G. E., 1999. Products of experts, 1–6.
- Hu, P., Wang, C., Li, B., Liu, M., 2012. Filling holes in triangular meshes in engineering. *Journal of Software* 7 (1).
- Lancaster, P., Salkauskas, K., 1981. Surfaces generated by moving least squares methods. *Mathematics of Computation* 37 (155), pp. 141–158.
- Li, Z., Meek, D., Walton, D., 2010. Polynomial blending in a mesh hole-filling application. *Journal of Computer-Aided Design* 42 (4), pp. 340–349.
- Pérez, E., Salamanca, S., Merchán, P., Adán, A., Cerrada, C., Cambero, I., 2008. A robust method for filling holes in 3d meshes based on image restoration. In: *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems. ACIVS '08*. Springer-Verlag, Berlin, Heidelberg, pp. 742–751.
- Pérez, E., Salamanca, S., Cerrada, C., Merchán, P., Adán, A., 2010. Revisión de técnicas de relleno de huecos en mallas 3d. In: *Acta de las XXX Jornadas de Automática*.
- Roth, S., Black, M. J., 2005. Fields of experts: A framework for learning image priors. In: *In IEEE Conference on Computer Vision and Pattern Recognition*. pp. 860–867.
- Salamanca, S., Adán, A., Cerrada, C., Adán, M., Merchán, P., Pérez, E., 2009. Reconocimiento de objetos de forma libre a partir de los datos de rango de una vista parcial usando cono curvaturas ponderadas. *RIAI*; Vol 4, No 1 (2007).
- Salamanca, S., Merchán, P., Adán, A., Pérez, E., Cerrada, C., 2008. Filling holes in 3d meshes using image restoration algorithms. In: *Proceedings of Fourth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 08)*.
- Sharf, A., Alexa, M., Cohen-Or, D., 2004. Context-based surface completion. *ACM Transactions on Graphics* 23 (3), pp. 878–887.
- Wang, J., Oliveira, M., 2007. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing* 25 (1), pp. 103–113.
- Wang, X., Liu, X., Lu, L., Li, B., Cao, J., Yin, B., Shi, X., 2012. Automatic hole-filling of cad models with feature-preserving. *Computers & Graphics* 36 (2), 101–110. [jce:title;Virtual Reality in Brazil 2011;ce:title;](#)
- Wei, M., Wu, J., Pang, M., 2010. An integrated approach to filling holes in meshes. In: *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence - Volume 03. AICI '10*. IEEE Computer Society, Washington, DC, USA, pp. 306–310.
- Wu, X., Wang, M., Han, B., 2008. An automatic hole-filling algorithm for polygon meshes. *Journal of Computer-Aided Design and Applications* 5 (6), pp. 889–899, cited By (since 1996) 1.