

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Ingeniería de Software y Sistemas Informáticos



**TÉCNICAS DE RELLENO DE HUECOS EN SUPERFICIES
ADQUIRIDAS MEDIANTE ESCÁNERES 3D**

TESIS DOCTORAL

Emiliano Pérez Hernández

Ingeniero en Organización Industrial

2011

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Ingeniería de Software y Sistemas Informáticos



TÉCNICAS DE RELLENO DE HUECOS EN SUPERFICIES ADQUIRIDAS MEDIANTE ESCÁNERES 3D

Memoria que presenta para optar al grado de Doctor

Emiliano Pérez Hernández

Ingeniero en Organización Industrial por la Universidad de Extremadura

Directores:

Santiago Salamanca Miño

Profesor Titular de Universidad del Departamento de Ingeniería Eléctrica, Electrónica y
Automática de la Universidad de Extremadura

Carlos Cerrada Somolinos

Catedrático de Universidad del Departamento de Ingeniería de Software y Sistemas Informáticos
de la Universidad Nacional de Educación a Distancia

Agradecimientos

Este trabajo ha sido parcialmente financiado por los Proyectos de Investigación del Ministerio de Educación y Ciencia con referencia DPI2005-03769, DPI2008-05444 y de la Junta de Extremadura PRI09C088 y de la CAM S2009/DPI-1559 (Robocity2030-II).

Agradecer esta tesis a mis directores Santiago Salamanca Miño y Carlos Cerrada Somolinos por depositar su confianza en mí desde mis comienzos en la investigación, y por brindarme la oportunidad de adentrarme en el apasionante mundo de la investigación 3D

Por otro lado, me gustaría dar las gracias a mis padres porque sin ellos no habría llegado a este punto, sin su apoyo, su esfuerzo y su cariño esto no habría sido posible. Además añadir mi agradecimiento a mis otros pilares imprescindibles: mis hermanos.

Por último, expresar mi agradecimiento a toda la gente que ha ido apareciendo a lo largo de este camino, y que se han convertido en grandes compañeros, amigos y mucho más que amigos. Gracias por vuestra paciencia y apoyo en los momentos que más lo he necesitado.

Gracias a todos.

Resumen

La digitalización de objetos para multitud de aplicaciones ha ido evolucionando de la mano de la evolución de los diferentes equipos de captura que han ido siendo desarrollados y puestos en el mercado. El proceso de digitalización puede dividirse en varias etapas, independientemente del sensor que se utilice. Este proceso parte de una etapa de adquisición, en la que se trabaja de forma manual con los equipos de captura. Le sigue una etapa de registro de todas las vistas parciales adquiridas en un sistema de referencia común. A continuación es necesario integrar todas esas vistas parciales en una única malla. Debido a la multitud de parámetros que intervienen en las etapas anteriores, siempre se incluye una etapa final de procesado que tratará de eliminar errores o pulir los resultados obtenidos. El procesado puede incluir una gran variedad de operaciones, dependiendo de las necesidades que vayan apareciendo. Entre estas operaciones son habituales las de suavizado de la malla; las de remallado o uniformización de la malla y una de gran importancia como es el relleno de huecos.

Los huecos aparecen en la malla debido a diversas razones tales como: oclusiones en la superficie del objeto, limitaciones en el posicionamiento del equipo de captura, superficies de baja reflectancia, errores en las etapas de registro e integración, etc. Se trata de una tarea que no es trivial, ya que pueden presentarse infinidad de casos de aparición de huecos, y, por tanto, no existe una solución que sea general a todos.

Por otro lado, existe una extensa investigación en el campo de los algoritmos de restauración de imágenes. La finalidad de estos algoritmos es la obtención de un proceso automático para la eliminación de elementos en una foto, como puede ser: puntos debido al ruido, suciedad en la imagen debido a la antigüedad de la película, texto impreso, objetos o personas, etc.

Puede encontrarse cierta similitud entre el problema que abordan los algoritmos de restauración de imágenes y el problema que se presenta en el relleno de huecos en superficies 3D: en ambos casos hay parte del conjunto de datos que se han eliminado y que necesitan ser restauradas.

Por tanto, la idea de esta tesis es abordar el diseño de un método de relleno de huecos en mallas 3D utilizando algoritmos de restauración de imágenes para reconstruir las partes que se han de completar. Para ello, se estudiará toda la problemática que ello conlleva, empezando por la primera diferencia que salta a la vista: se pretende aplicar métodos tradicionalmente aplicados en datos 2D (fotografías) a la resolución de un problema de datos 3D.

En primer lugar, se realizará un diseño inicial del método y su correspondiente análisis aplicándolo en vistas parciales obtenidas tras la digitalización. Tras la validación del método, se ha procedido al diseño de un método más general que aborde el problema en mallas completas que son obtenidas tras la digitalización y procesado de diversos tipos de objetos.

Puede concluirse, tras el análisis global de los resultados, que el método final se comporta de forma robusta en una gran variedad de situaciones, ofreciendo resultados de gran calidad para diversos tipos de huecos y diferentes tamaños de los mismos.

Abstract

The digitalization of objects for many applications has evolved hand in hand with the evolution of different capture devices which have been developed and put on the market. The scanning process can be divided into several stages, regardless of sensor used. This process begins by an acquisition stage, which is worked manually with the capture equipment. There follows an step of registration of all acquired partial views in a common reference system. Then it is needed to integrate all these partial views into a single mesh. Due to the many parameters involved in the earlier stages, it is always included a final stage of processing to try to debug or refine the results. Processing can include a wide variety of operations, depending on the needs that are appearing. These operations are commonly: smoothing the mesh, remeshing and an important one is the filling of holes.

Holes appear in the mesh due to various reasons such as occlusions on the surface of the object, limitations on the positioning of the capture equipment, low reflectance surfaces, errors in registration and integration stages, and so on. This is a task that is not trivial as they may occur many cases of appearance of holes, and therefore there is no general solution that is all.

On the other hand, there is extensive research in the field of image restoration algorithms. The purpose of these algorithms is to obtain an automatic process for removing items in a photo, such as: points due to noise, dirt in the image due to the age of the film, printed text, objects or people, etc. .

You can find some similarity between the problem that address image restoration algorithms and the problem that arises in filling gaps in 3D surfaces: in both cases a part of the data set have been removed and need to be restored.

Therefore, the idea of this thesis is to address the design of a method of filling holes in meshes using 3D image restoration algorithms to reconstruct the parts that have been completed. To this end, it will be studied all the problems that entails, beginning by the first difference which is obvious: it is intended to apply methods traditionally used in 2D data (photographs) to solving a problem of 3D data.

First, there will be an analysis of the problem and an initial design of appropriate method, applying it in partial views obtained after scanning. After validation of the method, we proceeded to design a general method to tackle the problem in whole mesh which are obtained after scanning and processing of various types of objects.

It can be concluded after a comprehensive analysis of the results, the final method behaves robustly in a variety of situations, offering high-quality results for different types of holes and sizes of them.

Índice general

1. Introducción	1
1.1. Motivo	1
1.2. Objetivos	3
1.3. Estructura de la tesis	4
1.4. Contribuciones de la tesis	5
2. Conceptos básicos de la creación de modelos 3D de objetos reales	8
2.1. Adquisición	10
2.1.1. Sistemas con contacto	11
2.1.2. Sistemas sin contacto	12
2.2. Registro	16
2.3. Representación de los datos e integración	17
2.3.1. Datos en bruto	18
2.3.2. Superficies	21
2.3.3. Sólidos	31
2.3.4. Estructuras de alto nivel	36
2.4. Procesado	37
2.5. Métodos de relleno de huecos.	40
2.5.1. Criterios de robustez del algoritmo de relleno de huecos	41
2.5.2. Clasificación de los métodos de relleno de huecos.	42
2.5.3. Revisión técnicas de relleno de huecos.	43
3. Verificación de los algoritmos de image inpainting en el rellenado de huecos	52
3.1. Introducción	52
3.2. Técnicas de restauración de imágenes	52
3.3. Método desarrollado. Aplicación de algoritmos de image inpainting al rellenado de huecos en mallas.	54
3.4. Generación de la imagen de rango	55
3.5. Aplicación de los algoritmos de restauración	62
3.6. Transformación de 2D a 3D. Triangulación de huecos	66
3.7. Integración de los huecos triangulados con la malla inicial	70
3.8. Validación del método de relleno propuesto	71
3.9. Resumen	88
3.10. Resultados experimentales	88
4. Definición de un método general para el rellenado de huecos en mallas 3D	89
4.1. Introducción	89
4.2. Descripción general del método para mallas completas	90
4.3. Identificación de los huecos	92

4.4.	Selección de la porción de malla	94
4.5.	Generación de la imagen de rango	95
4.5.1.	Plano de proyección sobre el que proyectar	95
4.5.2.	Resolución de la imagen de rango	104
4.6.	Aplicación de los algoritmos de image inpainting	107
4.7.	Integración de los resultados. Transformación de 2D a 3D. Triangulación de huecos	109
4.8.	Resultados experimentales	111
4.9.	Conclusiones	111
5.	Conclusiones	114
5.1.	Lineas de trabajo futuras	115
6.	Anexo I	117
7.	Anexo II	141
7.1.	El equipo de captura.	141
7.1.1.	Descripción específica del sensor de rango empleado.	141
7.2.	Descripción de los elementos y configuración empleados.	145
7.2.1.	El trípode y la cabeza de rotación	145
7.2.2.	La mesa o tabla de rotación. El patrón de calibración.	146
7.2.3.	La iluminación de la escena.	149
7.2.4.	El monitor de video.	151
7.2.5.	El ordenador.	153
7.3.	Estudio del brillo del objeto	154
7.4.	Estudio de la textura. Problemas ocasionados por los tonos oscuros.	156
7.5.	Detección y eliminación de puntos y caras erróneas.	156
7.5.1.	Tipos de corrección de errores aplicados:.	156
7.5.2.	Operaciones a realizar para corrección de la malla:	157
7.6.	Estudio de la ocupación óptima	161
7.7.	Generación de la imagen de rango sin rasterización	172
7.8.	Generación de la imagen de rango con rasterización	183
7.9.	Extensión de la porción de malla	187
7.10.	Resultados experimentales para mallas completas	198
7.11.	Estudio de la curvatura antes y después del relleno	201
8.	Referencias	207

Índice de figuras

2.1.	Imagen de una cabezal de digitalización en continuo SP 600 de Renishaw . . .	11
2.2.	Representación de un equipo de digitalización mediante brazo articulado. . .	12
2.3.	Izquierda: Cono de visión limitado por la silueta de la imagen y que contiene al objeto. Derecha: Intersección de varios conos de visión.	14
2.4.	Digitalización mediante escaner láser: (a) Barrido del laser sobre el objeto; (b) malla poligonal del objeto digitalizado; (c) representación suavizada de la malla poligonal; (d) inclusión de la textura en la representación del objeto virtual.	15
2.5.	Digitalización mediante la proyección de luz estructurada multicolor.	16
2.6.	Proceso de extracción de características en un objeto 3D.	16
2.7.	a) Fotografía de la escultura que va a digitalizarse; b) Nube de puntos estructurada; c) Nube de puntos desorganizada.	18
2.8.	Diferentes representaciones de un objeto: (a) fotografía para observar la información del color. (b) malla poligonal obtenida correspondiente a la digitalización de una vista parcial del objeto. (c) imagen de rango o niveles de profundidad correspondientes a esa vista parcial digitalizada.	19
2.9.	Representación del mapa de alturas correspondiente a una imagen de rango.	20
2.10.	Coordenadas cartesianas, esféricas y ejes ortogonales.	21
2.11.	Diferentes ejemplos de representación utilizando el método sopa de polígonos.	22
2.12.	Representación de una superficie paramétrica, con algunas curvas contenidas en la misma.[Rog01]	23
2.13.	(a) Superficie NURB con líneas de nudos (knot lines); (b) Red de puntos de control [Sev09].	24
2.14.	(a) Representación de la superficie implícita de un toro; (b) representación de la superficie "blobby" formada por la unión de dos esferas.	25
2.15.	Conversión de malla poligonal a superficie implícita [Tur01].	26
2.16.	Representación de una esfera mediante: (a) Malla estructurada; (b) Malla no estructurada.	27
2.17.	Representación de una malla hexagonal en 2D y su dual triangular [Mar07].	28
2.18.	Entidades que componen la malla triangular e interconexión entre ellas	29
2.19.	(a) Dos partes de la superficie confluyen en un vértice que produce una variedad no bidimensional (a la izquierda). Una arista que no produce una variedad bidimensional tiene más de dos caras incidentes (en el centro). La configuración de la derecha, aunque da lugar también a una variedad no bidimensional, es más habitual en muchas mallas [Bot06]; (b) Vecindad de un vértice dado v_i [Vas06].	31
2.20.	Descomposición de un objeto en diferentes celdas básicas.	33
2.21.	Representación mediante vóxeles de distinto tamaño de un mismo objeto.	33

2.22. (a) Esquema de partición del árbol octal u octree [Can02]; (b) y (c) ejemplos de representación de sólidos mediante árboles octales.	34
2.23. (a) Esquema de partición de un árbol BSP en el espacio 3D; (b) visualización de árbol BSP para un polígono en 2D, con su respectiva separación entre lo que queda dentro (<i>in</i>) y fuera (<i>out</i>) de cada partición.	35
2.24. Representación CSG: (a) Primitivas con sistemas de referencia; (b) Primitivas en un mismo sistema de referencia; (c) CSG resultante.	35
2.25. Modelado de sólidos mediante la técnica de barrido o <code>\it sweep</code>	36
2.26. Descomposición de un objeto en las diferentes partes que lo componen. El gráfico resultante formaría parte del grafo de la escena en donde se encuentre el objeto.	37
2.27. Ejemplo de extracción del esqueleto de objetos de formas diversas ([Cor07], [Pan10]).	38
2.28. Diversos tipos de errores habituales en mallas poligonales. [Ju09]	39
2.29. Proyección de huecos poligonales en un plano: (a) Hueco simple; (b) Hueco complejo [Jun05].	41
2.30. Ejemplo de relleno mediante funciones de base radial (RBF).(a) Malla con hueco. (b) Conjunto de centros utilizados para la interpolación. (c) Malla rellenada utilizando interpoladores RBF.[Bra06]	44
2.31. Ejemplo de relleno utilizando el método de funciones de base radial. (a) Malla con hueco. (b) Malla reparada. (c) Mapa del error entre la malla reparada y la original.[Wu08]	45
2.32. Ejemplo de malla con un hueco, y detección de las líneas de las características de la superficie.[Li10]	45
2.33. Ejemplo de relleno mediante desdoblado de la malla. (a) Hueco en la cabeza del modelo humano. (b) Malla desdoblada.(c)-(e) Malla final obtenida tras la minimización de energía.[Bru09]	46
2.34. Ejemplo de relleno mediante la técnica de subdivisión en huecos más simples [Li08]. En (a) y (b) mallas antes y después del relleno, respectivamente	47
2.35. Ejemplo de relleno de huecos complejos utilizando la técnica de cálculo de normales.[Zha07]	47
2.36. Ejemplo de relleno utilizando el método de ajuste de NURBs.[Kum07]	48
2.37. Interpolación de huecos grandes en dos imágenes de rango. (a) y (b) son dos imágenes de rango adquiridas de puntos de vista opuestos que ya están alineadas. (c) es una sección 2D del campo de distancias inicial, calculado a partir de las dos imágenes de rango. (d) - (h) resultados tras el relleno de huecos. (i) sección 2D de las mallas generadas.[Sag08]	49
2.38. Ejemplo de relleno de huecos utilizando la división en volúmenes atómicos propuesta en [Pod05].	49
2.39. Relleno de huecos con síntesis de textura.[Bre05]	51
2.40. Ejemplo de relleno de huecos utilizando un método basado en la información del contexto [Sha04].	51
3.1. (a), Restauración de una imagen antigua deteriorada; (b) Restauración de una imagen estropeada con ruido.	53
3.2. Etapas del método de relleno de huecos desarrollado	55
3.3. Elección del plano de proyección para hacer la transformación 3D a 2D: (a) elección errónea; (b) elección idónea.	56

3.4.	Dimensiones de la imagen de rango en función del tamaño del píxel elegido, q .	58
3.5.	Imágenes de rango en función del ancho de celda elegido	58
3.6.	Influencia de la uniformidad de la malla en la generación de la imagen de rango	59
3.7.	Relación entre el ratio de ocupación óptima O_{I_h} y el factor de proporcionalidad α	61
3.8.	Escalado de los valores, acercamiento en Z al plano $Z=0$	62
3.9.	(a) Imagen original; (b) Imagen con la máscara de selección aplicada, con la que se señala qué parte se desea eliminar; (c) Imagen resultante tras la aplicación del algoritmo de restauración propuesto por Criminisi et al. en [?].	63
3.10.	(a) Fotografía antigua deteriorada; (b) Restauración de la fotografía antigua mediante la aplicación del algoritmo propuesto en [?]; (c) Imagen original con un texto sobrepuesto que desea eliminarse; (d) Imagen con el texto eliminado tras la aplicación del algoritmo de [?].	65
3.11.	Ejemplo de Máscara de selección	66
3.12.	Obtención de los píxeles rellenos de la imagen restaurada mediante la aplicación de la máscara.	67
3.13.	Correspondencias entre las coordenadas 3D y las coordenadas del plano imagen.	68
3.14.	Puntos que se van a triangular (los de los huecos –en rojo- junto con los del borde de los mismo –en azul-): (a) Puntos del borde de los huecos; (b) Puntos que se van a triangular.	68
3.15.	Relacion de vecindad entre puntos / relación de vecindad entre pixeles. Nube de puntos estructurada	69
3.16.	Esquema de las triangulaciones boundary 1 U boundary 2	70
3.17.	Huecos una vez realizada la fase de triangulación.	71
3.18.	Malla final reconstruida, con los huecos coloreados en rojos para observar la reconstrucción	72
3.19.	Kd-tree, distancias entre dos mallas triangulares	74
3.20.	Comparativa de los algoritmos teniendo en cuenta la distancia media por hueco.	79
3.21.	Comparativa de los algoritmos teniendo en cuenta la distancia media por malla.	80
3.22.	Representación de la distancia media por malla para los tres algoritmos.	80
3.23.	Representación de la tasa de éxito frente a (a) el área total de los agujeros y (b) el área percentual. La línea continua es para el método propuesto, y la línea de puntos para el algoritmo comercial.	86
3.24.	Mallas (a) originales y (b) restauradas, digitalizadas de la colección del Museo Nacional de Arte Romano de Mérida.	87
4.1.	Etapas del método de relleno de huecos desarrollado	91
4.2.	Etapa inicial de identificación de los h_i ($i = 1, \dots, m$) huecos en la malla total M_T	94
4.3.	Procedimiento de selección de la porción de malla mediante la obtención de los anillos R_i , a partir del borde de hueco B_k	96
4.4.	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 v en cada caso: $v = 7$; $v = 5$; $v = 11$, respectivamente.	97
4.5.	Selección de diferentes planos de referencia o de proyección para cada uno de los huecos que se van a rellenar.	97
4.6.	Elección del plano de proyección para hacer la transformación 3D a 2D: (a) elección errónea; (b) elección idónea	98

4.7. (a), Anillo de triángulos que pertenecen al borde del hueco; (b) Normal al hueco obtenida como la media de las normales a las caras que son forman el borde de hueco; (c) Proyección del borde de hueco según la dirección de la normal al hueco.	99
4.8. Three subfigures.	100
4.9. Cálculo de la orientación de ligado al plano de referencia, que haga que el Área de P_k , A_k , sea máxima	101
4.10. En (a) se muestra la representación 3D a la izquierda y la proyección sobre el plano de referencia inicial, del hueco 1. En (b) se muestra la representación 3D a la izquierda y la proyección sobre el plano de referencia que se ha obtenido tras aplicar el algoritmo de maximización del area.	102
4.11. Posición del plano de referencia con respecto a la porción de malla.	104
4.12. Aparición de pérdidas en la imagen de rango debidas a la orientación del plano de referencia con respecto a la malla.	106
4.13. Aparición de más de un hueco en la imagen de rango de uno de ellos	107
4.14. Representación de la 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.	109
4.15. Integración de las zonas que se han rellenado aplicando el algoritmo de restauración en la malla total M_T de la figura 4.2. Huecos 1, 2 y 3 en (a), (b) y (c), respectivamente.	112
7.1. Vista frontal y trasera y diagrama de medidas del sensor Vivid 300 de Minolta.	142
7.2. Principio de medida del sensor de rango.	143
7.3. Imagen del trípode Manfrotto 475 utilizado como soporte del escáner.	146
7.4. Imagen de la cabeza de rotación Manfrotto 279 utilizada para la orientación del escáner.	147
7.5. Imagen del conjunto formado por trípode, la cabeza de rotación y el escáner.	147
7.6. Imagen de la mesa de rotación situada en el entorno de captura con iluminación controlada	148
7.7. Controlador de la mesa de rotación de la marca ISEL	148
7.8. Imagen del patrón de calibración centrado sobre el plato giratorio	149
7.9. Flexos utilizados para el control de iluminación de la escena.	150
7.10. Entorno de captura con iluminación controlada	150
7.11. Monitor de video Sony PVM-14L1 utilizado para visualizar la señal de video del escaner	152
7.12. Controles accesibles en el frontal del monitor y conexiones de entrada y salida disponibles en la parte trasera del mismo	152
7.13. (a). Imagen captada por la cámara del escáner que es mostrada en el monitor de video;(b). Imagen a la que realmente se le aplica el proceso de triangulación	153
7.14. Ordenador que se ha utilizado para el sistema de adquisición de superficies .	154
7.15. Esquema general que muestra las interconexiones entre los elementos que componen el sistema	155
7.16. Ejemplo de caras con topología Non-manifold.	157
7.17. Ejemplo en donde la cara (V1 V2 V3) es una cara redundante.	158
7.18. Ejemplo de caras cruzada en donde la arista (V2 V3) de la cara (V1 V2 V3) está cruzada con la arista (V1 V4)	158
7.19. Ejemplo de caras inestables en la modalidad de normales erróneas	158

7.20. Ejemplo de caras inestables en la modalidad de caras giradas	159
7.21. Ejemplo de dos caras que intersectan entre sí.	159
7.22. Ejemplo de “folded faces”, es decir, caras adyacentes que comparten una arista y con un ángulo entre ellas muy pequeño.	160
7.23. Figura 123.- (a) y (b) Estado anterior y posterior, respectivamente, al cosido de bordes.	161

Índice de tablas

2.1.	Nueve criterios para elegir un equipo de digitalización adecuado	10
3.1.	Selección de un subconjunto de imágenes del banco de imágenes de Berkeley utilizado para el entrenamiento del algoritmo propuesto por Roth et al [?] . .	65
3.2.	Distancia media por malla en función del algoritmo de image inpainting . . .	75
3.3.	Distancia media por malla en función del algoritmo de image inpainting . . .	76
3.4.	Representación de la distancia media por hueco en cada malla	81
6.1.	Generación de huecos artificiales en 20 vistas parciales de la base de datos .	118
6.2.	Comparación de algunas mallas rellenas con cada uno de los tres algoritmos aplicados	125
6.3.	Valor máximo y mínimo de la distancia entre nodos para los mapas de distancias	133
6.4.	Mapas de distancias entre las mallas originales y las mallas reconstruidas . .	134
7.1.	Estudio de la ocupación óptima	162
7.2.	Generación de la imagen de rango sin rasterización	173
7.3.	Generación de la imagen de rango con rasterización	184
7.4.	Extensión de la porción de malla	188
7.5.	Resultados experimentales para mallas completas	199
7.6.	Estudio de la curvatura antes y después del relleno	202

Capítulo 1

Introducción

1.1. Motivo

Prácticamente desde los inicios de su historia, el hombre, haciendo uso de su inteligencia, ha tratado de representar el entorno que le rodea para los fines más diversos: expresión artística, representaciones religiosas, esquemas de fabricación, planificación bélica, transmisión y almacenamiento de conocimientos o ideas, etc. Para tales representaciones, ha empleado multitud de técnicas a lo largo de la historia, tales como: pintura, escultura, dibujo, maquetas, etc. En una posible distribución cronográfica de esas técnicas, saltaría a la vista la inflexión que supuso la irrupción de los ordenadores en el siglo XX, lo que da lugar a una revolución en los métodos de representación y modelado existentes hasta el momento.

Los ordenadores se han convertido en una herramienta, actualmente imprescindible, en el campo de la representación y modelado, principalmente en lo relativo a la fabricación y construcción, en donde, permiten un estudio más preciso y detallado, así como una simulación del comportamiento en diversos escenarios, de la entidad que se desea fabricar o construir. Además, la posibilidad de creación de mundos virtuales, recreados en un ordenador, en donde experimentar, estudiar, recrear, divertirse, suponen un potencial de infinitas posibilidades. Ahondando aun más en las aplicaciones relacionadas con la fabricación, el modelado CAD y la fabricación asistida CAD/CAM ha permitido un cambio cuantitativo y cualitativo colosal en la producción industrial. Y además, se han añadido posibilidades muy interesantes tales como la ingeniería inversa, lo que permite, partiendo de un objeto fabricado o real, volver hacia atrás en su esquema de fabricación hasta llegar al concepto, la idea, el modelo inicial, el cual puede ser inyectado de nuevo en la cadena de producción, con las mismas características o con características mejoradas.

Los métodos para la representación o modelado de objetos reales por ordenador, han ido evolucionando a lo largo del tiempo. Así, inicialmente solo se podían introducir en el ordenador, de forma relativamente manual, las medidas tomadas en el mundo real, mediante los métodos de medición existentes. Las técnicas que supusieron un gran cambio fueron las que aplicaban una serie de sensores específicos que permitían capturar la realidad y enviarlas de forma automática al ordenador, tales como: sensores de rango o escáneres 3D. El uso de esas técnicas se ha extendido a multitud y variados campos como: fabricación, ingeniería inversa, reconocimiento, conservación del patrimonio, bases de datos de objetos, creación de mundos virtuales con elementos reales, efectos especiales, etc.

Dentro de ese contexto, antes del planteamiento de esta tesis, en el grupo al que pertenezco se estaba desarrollando el proyecto *Diseño, desarrollo y aplicación de técnicas de reconstrucción y visualización de escenas 3d complejas mediante sensores de rango*, en el que se

definían como objetivos la construcción de un sistema de reconocimiento de escenas 3D formado por un sensor de rango, un brazo robot y una línea de montaje sobre la que pasarían los conjuntos objetos a reconocer. En ese proyecto se desarrolló un algoritmo de segmentación y reconocimiento, con el que se obtuvieron unas tasas de éxito elevadas. La idea era que tras el proceso de reconocimiento se le mandaban al robot que coge las piezas, las consignas adecuadas para que fuese adaptando sus movimientos según las necesidades requeridas por la escena, ya conocida. Para realizar el reconocimiento era necesario disponer de una base de datos 3D de todas las posibles objetos a reconocer. La creación de esa base de datos fue llevada a cabo mediante el empleo del adecuado equipo de captura. Esta tarea consistió en mi principal contribución al desarrollo de dicho proyecto.

El inconveniente que se comprobó que presentaban estos procedimientos de reconocimiento es que se basan en la búsqueda y comparación de valores dentro de bases de datos de objetos. Por tanto, si ya de por sí son algoritmos muy costosos computacionalmente, a medida que el tamaño de la base de datos aumenta, el tiempo de cálculo alcanza valores muy elevados. Por esto, se hace muy difícil la aplicación de estos métodos en sistemas que requieran un reconocimiento en tiempo real.

Seguidamente se continuó con el proyecto *Integración de sensores visuales de rango y de radiofrecuencia aplicada a tareas de reconocimiento de objetos en entornos de fabricación automatizada* que añadía como elemento innovador al sistema anterior la incorporación de etiquetas y sensores RFID. Con esto se conseguía que el sistema descrito anteriormente fuese asistido por esta nueva tecnología y se consiguiesen unos tiempos de proceso inferiores a los anteriores.

Este proyecto dio lugar a mis primeros trabajos de investigación dentro del periodo de investigación de mis cursos de doctorado. En concreto, consistieron en el análisis global de los resultados y en la obtención de medidas cuantitativas de la mejora que suponía este último proyecto con respecto al anterior. Tras ese análisis se concluyó que la hipótesis por la que se planteaba la introducción del RFID, reducción de tiempos de ejecución y mejoras de las tasas de reconocimiento, podía ser validada.

Sin embargo también se observó que la mejora era menor de la que cabría esperar en un principio. Por ello se decidió intentar mejorar el método, comenzando en primer lugar por la modificación del algoritmo de reconocimiento.

El algoritmo de reconocimiento estaba basado en el uso de modelos esféricos de mallas. Estos modelos, si se estudian a fondo, a pesar de ofrecer buenos resultados, presentan muchas limitaciones en cuanto a la forma y constitución de los objetos y escenas con que se trabaje.

Por tanto, se planteó como siguiente objetivo el planteamiento de nuevos modelos de mallas que no fuesen esféricos. Y, a partir de ahí, se decidió introducirse en el campo del modelado por mallas triangulares, comenzando por el estudio de las técnicas de reconstrucción de superficies 3D de objetos reales.

La creación de modelos 3D de objetos reales con cierta precisión no es una tarea trivial. Además, para llevarla a cabo no son válidas las técnicas tradicionales de modelado. En tales situaciones, se hace imprescindible el uso de sensores de rango o escáneres 3D debido a su relativa independencia de la geometría adquirida y al corto tiempo de adquisición.

La reconstrucción completa de la superficie 3D a partir de la información adquirida con un escáner 3D puede dividirse en varias etapas. Una primera etapa de registro de las diferentes vistas parciales, en la que se calcula un sistema de referencia común para todas ellas. La siguiente etapa es la integración de las vistas parciales con el fin de obtener una única representación matemática de la superficie. En la siguiente etapa se procesará la superficie 3D obtenida. Este procesado puede incluir operaciones de suavizado, remallado, detección y

eliminación de elementos erróneos, rellenado de huecos, etc.

Cuando una malla se encuentra cerrada, suponiendo que no existen caras o bordes mal definidos. Existen multitud de aplicaciones que necesitan que la malla sea hermética: sistemas de prototipado rápido, sistemas de metrología, modelos 3D para aplicaciones de visión por computador o para graficas por computador, etc. La razón fundamental de que la malla no sea hermética es la aparición de huecos en ella. Es, por tanto, de vital importancia la fase de rellenado de huecos para así conseguir una malla cerrada.

Existen dos hechos que hacen que aparezcan huecos en una malla. El primero es que en el proceso de registro y/o integración se haya producido alguna pérdida o redefinición incorrecta de la malla, por ejemplo, algún triángulo que falte por definir. Este tipo de huecos no suponen un problema y suelen rellenarse de forma directa sin tener en cuenta la información geométrica que rodea al hueco. El segundo hecho es la falta de información geométrica debido a oclusiones o sombras en los objetos. En este caso los huecos suelen ser de mayor tamaño que los anteriores. Además, cuando se trabaja con objetos de forma libre, con cierto nivel de detalle, pueden encontrarse huecos de cuyas superficies podrían ser muy complejas (por ejemplo, con grandes variaciones en la curvatura), y su rellenado se convierte en un problema de gran dificultad. Es por esto que, aunque la búsqueda de una solución haya sido abordada por muchos autores, no existe aún un procedimiento general para todas las posibles situaciones que se pueden presentar. Por ello se requiere normalmente de la intervención del usuario para poder evaluar el problema y no existe un algoritmo automático que le dé solución. Dentro del proceso de reconstrucción 3D, el relleno de huecos es en la actualidad un tema suficientemente abierto, en el que todavía el factor humano es determinante y que despierta un gran interés desde un punto de vista investigador. Todo ello justificaba que se plantease el desarrollo de esta tesis con el objetivo de la resolución de este problema.

1.2. Objetivos

En esta tesis se aborda por tanto el problema del relleno de huecos. Para su abordaje y resolución se plantea el desarrollo de diferentes técnicas con las que se pretende alcanzar los siguientes objetivos principales:

- Que permitan el funcionamiento de forma automática.
- Que sean válidas para diversas situaciones y tipos de datos.

Para tal fin se establece una metodología de trabajo estructurada en los siguientes puntos:

1. Documentación bibliográfica: Recopilación de textos, artículos y referencias en general especializados en los temas propuestos.
2. Definición del/de los algoritmo/s de rellenado: Dado que se observa que los algoritmos de reparación de imágenes son similares a los de rellenado y son bastante robustos, se plantea su aplicación en el relleno de huecos en objetos 3D.
3. Adaptación del/de los algoritmo/s para trabajar con objetos 3D: Para ello es necesario adaptar la forma de representación de los datos 3D para que sean entendibles por esos algoritmos. Ya se ha trabajado en este aspecto, proponiendo un método basado en la proyección de los datos 3D sobre un plano y su almacenamiento en cuadrículas o píxeles de un ancho determinado. Lo que así se obtiene, con la correspondiente normalización,

es una imagen en niveles de grises que almacena la información de la superficie 3D desde un punto de vista determinado (imagen de rango).

Otro factor que hay que estudiar es el tamaño de la imagen de rango, en número de píxeles, en función de la resolución de la malla con que se está trabajando. Habrá que determinar un valor experimental óptimo que dé lugar a una imagen de rango que represente adecuadamente la información 3D de la vista parcial.

4. Prueba del/de los algoritmo/s sobre vistas parciales con huecos generados artificialmente: En primer lugar se reunirá un conjunto de vistas parciales sobre las que probar los algoritmos. En este caso particular, el problema de la elección del eje de proyección a partir del que generar la imagen de rango está resuelto.

Los huecos, que se generen artificialmente, tendrán formas y tamaños variados. Los resultados podrán ser comparados con las vistas parciales, lo que proporcionará una medida de cuán eficaz y robusto es el método que se proponga.

5. Adaptación del/de los algoritmo/s para modelos completos: En este punto se pretende conseguir un método de relleno automático de huecos en mallas completas. Cada una de estas mallas es obtenida tras la digitalización y fusión de las diferentes vistas de un objeto real. Pueden existir diversas causas para la aparición de huecos, tales como: auto-oclusiones debido a la forma del objeto; brillos en la superficie con la consiguiente pérdida de información; limitaciones en el posicionamiento del escáner con la consiguiente falta de datos reales, etc. Para plantear el rellenado en estas situaciones, se habrán de estudiar las siguientes particularidades:

- Cómo efectuar la proyección de cada zona a rellenar sobre un plano.
- Qué tamaño habrá de tener la imagen con respecto al tamaño del objeto.
- Qué resolución habrá de tener la imagen de rango.
- En qué orden proceder al rellenado, cuando se presentan varios huecos.
- Qué limitaciones presenta este método, en cuanto a tamaño del hueco, concavidad o convexidad de la superficie.etc.

En el estudio de estas situaciones se utilizarán datos 3D procedentes de la digitalización de esculturas ubicadas en el Museo Nacional de Arte Romano de Mérida.

1.3. Estructura de la tesis

El trabajo desarrollado en la tesis se va a vertebrar utilizando 3 capítulos y 3 anexos. En primer lugar, en el capítulo 2 se va a contextualizar el problema tratado en la tesis, haciendo un repaso por los conceptos básicos de la creación de modelos 3D de objetos reales. En este capítulo se comenzará tratando las etapas básicas que componen dicho proceso de creación (adquisición, registro, representación, integración, procesado), y se acabará con una explicación detallada del problema que se aborda en la tesis: el relleno de huecos. Además de el análisis inicial del problema se expondrá una extensa revisión de las diversas técnicas de relleno de huecos que pueden encontrarse al sondear el estado actual del arte de esas técnicas.

En el capítulo 3 se comenzará, tras un primer planteamiento del método ideado de relleno, con una revisión de técnicas de restauración de imágenes disponibles en la literatura

especializada. Seguidamente, se propone la verificación del método desarrollado inicialmente aplicándolo únicamente a vistas parciales de objetos. Para ello, van a desarrollarse cada una de las etapas que componen dicho método en las diferentes secciones del capítulo. Finalmente, se tratará de efectuar las pertinentes mediciones que demuestren si puede ser validada la técnica de relleno que se ha ideado. Para terminar, se expondrán los diferentes resultados globales que se han alcanzado para vistas parciales.

Una vez validado el método, en el capítulo 4, se propone su adaptación para que sea aplicado en en mallas 3D completas. El nuevo método se va a estructurar prácticamente en las mismas etapas que en el capítulo anterior. Por tanto, el capítulo se dividirá en las mismas etapas que en el caso de vistas parciales, pero en cada una de ellas se irán explicando las pertinentes modificaciones que han de aplicarse por las particularidades que confieren el uso de mallas 3D completas. Al final del capítulo se expondrán también los resultados globales que se han obtenido y las conclusiones que de ellos se extraen.

Para concluir la tesis se enumerarán las conclusiones globales que han emanado a lo largo del desarrollo del trabajo de la tesis y se lanzarán una serie de sugerencias e ideas sobre trabajos futuros, algunos ya comenzados y otros que quedan en el tintero para continuar trabajando en la resolución del problema de la tesis o bien para extender la idea planteada a otros campos de aplicación.

Al final de la memoria se incluyen 3 anexos que en los que se expondrán una larga lista de figuras y gráficos con los resultados particulares obtenidos tanto en la aplicación con vistas parciales como con objetos completos, así como una exposición de las principales partes del código desarrollado, respectivamente.

1.4. Contribuciones de la tesis

Los trabajos desarrollados desde el inicio del periodo de doctorado hasta la redacción final de esta tesis, han dado lugar a diversas publicaciones y asistencias a congresos, que se enumeran a continuación:

■ Publicaciones (libros):

- Carlos Cerrada, Santiago Salamanca, Emiliano Pérez, José A. Cerrada e Ismael Abad, *Fusion in 3D Vision Techniques and RFID Technology for Object Recognition in Complex Scenes*, Proceedings del IEEE International Symposium on Intelligent Signal Processing, 6 páginas, ISBN: 1-4244-0829-6.
- Emiliano Pérez Hernández, Santiago Salamanca Miño, Antonio Adán Oliver, Carlos Cerrada Somolinos, Pilar Merchán García, Inés Tejado Balsera, Aplicación de algoritmos de reparación de imágenes al relleno de huecos en mallas 3D, Acta de las XXVIII Jornadas de Automática, 9 páginas, ISBN: 978-84-690-7497-8.
- Vicente Domínguez González, Apolonio Oliver del Hierro, Antonio Adán Oliver, Santiago Salamanca Miño, Emiliano Pérez Hernández, Carlos Cerrada Somolinos, Creación de modelos completos 3D sobre piezas escultóricas, Acta de las XXVII Jornadas de Automática, 9 páginas, ISBN: 978-84-690-7497-8.

■ Publicaciones (revistas):

- Emiliano Pérez, Santiago Salamanca, Pilar Merchán, Antonio Adán, Carlos Cerrada, Inocente Cambero, A Robust Method for Filling Holes in 3D based on Image Restoration, *Advanced Concepts for Intelligent Vision Systems 2008*, Lecture Notes in Computer Science, Juan-LesPins, Francia, Octubre 2008.
 - Carlos Cerrada, Santiago Salamanca, Emiliano Pérez, J. A. Cerrada, Ismael Abad, Improved Method for Object Recognition in Complex Scenes by Fusing 3D Information and RFID Technology: Performance Analysis, *IEEE Transaction on Instrumentation and Measurement (ISSN: 0018-9456)*, Vol. 58, Num:10, pp. 3473-3480, 2008.
 - Santiago Salamanca, Antonio Adán, Carlos Cerrada, Miguel Adán, Pilar Merchán, Emiliano Pérez, Reconocimientos de Objetos de Forma Libre a Partir de los Datos de Rango de una Vista Parcial Usando Cono Curvaturas Ponderadas, *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*, Vol. 4, pp. 95-106, 2007.
- Asistencias a congresos:
 - E.Pérez, S.Salamanca, C.Cerrada, P. Merchán, A. Adán, Técnica de relleno de huecos en mallas 3D completas mediante algoritmos de restauración de imagen, *Simposio CEA de Visión por computador y modelado 3D*, Badajoz (Spain), Julio 2011.
 - E.Pérez, S.Salamanca, C.Cerrada, Un método robusto para relleno de huecos en mallas 3D basado en algoritmos de restauración de imagen, *7th Workshop Robocity 2030*, Madrid (Spain), Octubre 2010
 - Emiliano Pérez Hernández, Santiago Salamanca Miño, Carlos Cerrada Somolinos, Pilar Merchán García, Antonio Adán Oliver, Revisión de Técnicas de Relleno de Huecos en Mallas 3D, *XXXI Jornadas de Automática*, Jaén (Spain), Septiembre 2010.
 - A. Adán, S. Salamanca, P. Merchán, T. Nogales, V. Domínguez, R. Chacón and E. Pérez, Digitalization and 3D Reconstruction of Aeneas Group, *The 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2008)*, The Eurographics Association 2008, Braga (Portugal), Diciembre 2008.
 - A. Martin, E. Perez, S. Salamanca, P. Merchan and A. Adan, Heritage pieces integration in autonomous augmented reality systems: key problems and solutions, *The 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2008)*, The Eurographics Association 2008, Braga (Portugal), Diciembre 2008.
 - Emiliano Pérez, Santiago Salamanca, Pilar Merchán, Antonio Adán, Carlos Cerrada, Inocente Cambero, A Robust Method for Filling Holes in 3D based on Image Restoration, *Advanced Concepts for Intelligent Vision Systems 2008*, Juan-LesPins (Francia), Octubre 2008.
 - Santiago Salamanca, Pilar Merchán, Emiliano Pérez, Antonio Adán, Carlos Cerrada, Filling Holes in 3D Meshes using Image Restoration Algorithms, *Four International Symposium on 3D Data Processing, Visualization and Transmission*, Atlanta (Georgia,EEUU), Junio 2008.

- Carlos Cerrada, Santiago Salamanca, Emiliano Pérez, José A. Cerrada e Ismael Abad, Fusion in 3D Vision Techniques and RFID Technology for Object Recognition in Complex Scenes, IEEE International Symposium on Intelligent Signal Processing, Alcalá de Henares (España), Octubre 2007.
- Emiliano Pérez Hernández, Santiago Salamanca Miño, Antonio Adán Oliver, Carlos Cerrada Somolinos, Pilar Merchán García, Inés Tejado Balsera, Aplicación de algoritmos de reparación de imágenes al rellenado de huecos en mallas 3D, XXVIII Jornadas de Automática, Huelva (España), Septiembre de 2007.
- Vicente Domínguez González, Apolonio Oliver del Hierro, Antonio Adán Oliver, Santiago Salamanca Miño, Emiliano Pérez Hernández, Carlos Cerrada Somolinos, Creación de modelos completos 3D sobre piezas escultóricas, XXVIII Jornadas de Automática, Huelva (España), Septiembre de 2007.

Capítulo 2

Conceptos básicos de la creación de modelos 3D de objetos reales

Dentro del ámbito del que trata esta tesis, reconstruir un objeto tridimensional, se refiere a obtener una representación matemática del mismo para poder ser almacenado, visualizado o procesado. Los métodos para la reconstrucción de objetos tridimensionales han incrementado su velocidad y precisión considerablemente en los últimos años gracias a diversos avances científicos, tales como el desarrollo de láseres, chips CCD, y el aumento de velocidad de trabajo de los procesadores.

La creación de modelos tridimensionales es de utilidad para un amplio rango de aplicaciones tales como fabricación, simulación virtual, exploración con fines científicos, medicina, marketing, etc. Se exponen a continuación algunas de las más relevantes [Cur97]:

- Visión por computador: Las técnicas de visión por computador tienen como fin último extraer propiedades del mundo a partir de diversas fuentes de información tales como: una imagen o conjunto de imágenes, secuencias de video, digitalizadores o escáneres, etc. Existen multitud de aplicaciones de la visión artificial, entre las que cabe destacar las siguientes: control de procesos (p. ej.: operación automática de robots industriales, guiado de vehículos autónomos); evaluación automática de la calidad de piezas; detección de eventos (p. ej.: vigilancia o conteo de personas); modelado de objetos o entornos (p. ej.: análisis de imágenes médicas o modelado topográfico); diseño de dispositivos de interacción humana con las máquinas, etc.
- Gráficas por ordenador: En ellos se incluyen las técnicas desarrolladas para generar imágenes visuales sintéticamente e integrar o cambiar la información visual y espacial adquirida del mundo real. Actualmente los gráficos por ordenador se aplican en multitud de aspectos de la vida cotidiana tales como información sobre la predicción del tiempo, investigaciones médicas, efectos especiales en cine, creación de productos para la industria del videojuego, realidad virtual, etc.
- Ingeniería inversa: Algunos componentes para la fabricación de un producto se diseñan actualmente mediante sistemas CAD. Sin embargo, en algunos casos, se dispone de un componente mecánico del que no existe un modelo 3D previo para poder regenerarlo. Esta situación se da habitualmente en componentes de máquinas que fueron diseñadas antes del uso sistemático de CAD en fabricación, así como en componentes que fueron ajustados manualmente para que encajasen en algún sistema. Si alguno de estos componentes se deteriora y no existen repuestos, puede ser útil la digitalización de la pieza de forma precisa para poder fabricarlo de nuevo.

- *Inspección*: Una vez que un fabricante ha creado un modelo 3D en el ordenador de un componente mediante digitalización de un objeto real o mediante diseño CAD, dispone de un prototipo para comenzar la fabricación en cadena del producto. En determinadas tipos de componentes las dimensiones del producto a fabricar solo acepta determinadas tolerancias con respecto al modelo CAD. En este caso, la digitalización de la pieza puede ser de utilidad para determinar qué zona convendría modificar del modelo CAD para que el proceso de fabricación produzca resultados aceptables.
- *Estudio y conservación de piezas del Patrimonio Cultural*: Las piezas de museo representan un tipo de objetos que son de interés para diversos colectivos: historiadores, científicos, educadores, etc. Tradicionalmente, para observar estos objetos se ha hecho necesario visitar físicamente el museo u obtener imágenes o secuencias de video que no ofrecen ninguna interactividad. Mediante la digitalización de estas piezas puede conseguirse que multitud de valiosas esculturas estén disponibles para su visualización y estudio. Para los científicos, estos modelos virtuales ofrecen la oportunidad para estudiar y medir determinados parámetros de forma remota y mediante el uso de potentes herramientas digitales.
- *Medicina*: Dentro de la medicina existen multitud de aplicaciones potenciales de la digitalización de piezas. Así, por ejemplo, en el diseño de prótesis se requiere ajustar las dimensiones de una pieza a un paciente determinado con alta precisión. Los cirujanos plásticos pueden utilizar el cuerpo digitalizado de un paciente para simular y visualizar los resultados de una cirugía. En los tratamientos de radioterapia, puede ser de utilidad el uso de un modelo del paciente para guiar al médico en la dirección de aplicación de radiación de forma precisa.
- *Otras aplicaciones*: compras a distancia, al permitir añadir modelos 3D en los catálogos de productos, con las posibilidades exploración que ello ofrece; o el diseño, revisión y corrección de prototipos de productos.

Para capturar un objeto completo se ha de realizar un muestreo de un conjunto de puntos discretos de su superficie. Como resultado se obtiene un volumen de datos que debe ser eficientemente tratado con determinados algoritmos para generar un modelo 3D con suficiente precisión.

Actualmente, gracias al desarrollo de técnicas de digitalización y al abaratamiento del hardware especializado, se ha extendido el uso de sensores de rango o escáneres 3D debido a su relativa independencia de la geometría adquirida y al corto tiempo de adquisición.

La reconstrucción completa de la superficie 3D a partir de la información adquirida con un escáner 3D puede dividirse en las siguientes etapas:

- *Adquisición*: Por razones obvias, cuando se adquiere la superficie de un objeto, se dispone de una sola vista parcial. Para obtener la superficie completa habrá que adquirir varias vistas parciales.
- *Registro*: Se realiza la alineación de las diferentes vistas parciales, y se calcula un sistema de referencia común para todas ellas.
- *Integración*: Se obtiene una única representación de toda la superficie, sin distinguir entre las vistas parciales.

- *Procesado*: En esta etapa se incluyen operaciones tales como suavizado, remallado, detección y eliminación de elementos erróneos, relleno de huecos, etc.

A lo largo de este capítulo se van a explicar cada una de estas fases de forma más detallada. De esta forma, en la sección 2.1, se enumerarán y clasificarán diversos sistemas de digitalización que se encuentran disponibles para efectuar la primera etapa; en la sección 2.2 se describirán brevemente las técnicas para el registro de vistas parciales; en la sección ?? se analizarán los modelos de representación de superficies más usuales, centrándonos en el que se ha utilizado particularmente, en la presente tesis, el modelo de malla triangular. A continuación, en la sección 2.4 se describirán las operaciones más comunes que se aplican en la etapa de procesado, para finalizar en la sección 2.5 con un exhaustivo análisis de las técnicas de relleno de huecos, que es el principal objeto de estudio de esta tesis.

2.1. Adquisicion

Como se mencionó en el apartado anterior, el hardware para la digitalización 3D ha evolucionado considerablemente en los últimos años, aumentando la velocidad y precisión de digitalización y disminuyendo los costes. En el mercado se encuentran disponibles diferentes tipos de escaneres 3D cada uno de ellos con unas características específicas. En [Pav07] proponen una lista de nueve criterios a tener en cuenta a la hora de elegir un sistema de digitalización 3D (ver tabla 2.1).

Tabla 2.1: Nueve criterios para elegir un equipo de digitalizacion adecuado

Nº	Criterio
1	Coste
2	Material a digitalizar
3	Tamaño del objeto a digitalizar
4	Equipo transportable
5	Precisión
6	Adquisición de textura
7	Productividad de la técnica
8	Necesidad de personal especializado
9	Adaptación de los datos generados a los estandares

Una posible clasificación de los sistemas de digitalización 3D sería teniendo en cuenta si hay o no contacto con la superficie a adquirir, y de esta forma tendríamos los siguientes grupos:

- Sistemas con contacto
- Sistemas sin contacto

En los siguientes puntos se describirán brevemente algunos de los métodos disponibles dentro de cada uno de los grupos mencionados.



Figura 2.1: Imagen de una cabezal de digitalización en continuo SP 600 de Renishaw

2.1.1. Sistemas con contacto

Estos sistemas de digitalización 3D son los primeros que surgieron. Mediante ellos se obtienen las coordenadas 3D de los puntos de una superficie gracias al desplazamiento de una punta sobre la misma. El primer inconveniente que salta a la vista es la lentitud con que se adquiere la superficie de un objeto. Además, para emplear estos sistemas por contacto, se necesita que las piezas tengan la rigidez suficiente para que no se deformen por el contacto de la punta. Por otra parte, debido a la geometría de las puntas, normalmente esférica, es imposible digitalizar algunas ranuras y ángulos interiores.

En la actualidad existen cabezales de digitalización en continuo, como el que se muestra en la figura 2.1, que permiten una velocidad de adquisición mayor respecto a los convencionales, al no tener que separarse de la superficie a digitalizar.

Otro tipo de sistema de digitalización 3D con contacto son los brazos articulados de operación manual, como el mostrado en la figura 2.2. Estos poseen una elevada precisión; pero por el contrario tienen una velocidad de adquisición de datos muy baja, ya que la punta necesita ser llevada manualmente cada punto que se quiera digitalizar, con lo cual la digitalización de un objeto de tamaño medio resulta muy costosa.

En cuanto a su aplicación, principalmente se utilizan en la verificación dimensional de piezas industriales (control de calidad).



Figura 2.2: Representación de un equipo de digitalización mediante brazo articulado.

2.1.2. Sistemas sin contacto

La principal ventaja de los digitalizadores 3D sin contacto es que tienen una velocidad de adquisición de datos muy superior a las de los que requieren contacto. Una posible clasificación de este tipo de digitalizadores es la basada en la técnica de visión que utilicen. De esta forma se tendrán los siguientes grupos, que serán explicados a continuación:

- Técnicas de visión pasiva
- Técnicas de visión activa

Técnicas de visión pasiva

El sistema visual humano permite obtener información de profundidad mediante la fusión de dos escenas monoculares, que son las escenas que captan cada uno de los ojos. Este sistema permite al hombre “ver en 3D” y es conocido como principio de visión estereoscópica. Así pues, dicho principio se basa en utilizar dos puntos de vista de un mismo objeto para encontrar las coordenadas tridimensionales. Para determinar la posición de un punto a partir de dos imágenes es necesario tener un modelo del sistema óptico utilizado.

Este principio general puede mejorarse con modelos de cámaras más elaborados o utilizando más de dos cámaras, lo que se conoce con el nombre de fotogrametría. La visión estereoscópica presenta el interés de poder dar la posición en 3D de puntos sobre las superficies independientemente de la iluminación específica (visión pasiva).

Estereoscopios El objetivo de este método es la extrapolación de la información geométrica de un objeto a partir de un par de fotografías tomadas desde ángulos y posiciones relativas conocidas, simulando la visión humana. Estos sistemas tienen gran importancia en el campo de la robótica y la visión por ordenador. Como ventajas han de resaltarse la posibilidad de captura de geometría y textura simultáneamente, el bajo coste y la transportabilidad. Por contra, las desventajas del sistema son la baja resolución que ofrecen y la necesidad de una calibración adecuada para su correcto funcionamiento.



Digitalización mediante video Esta es una variante del estereoscopio, pero en lugar de dos cámaras se utiliza una cámara de video que captura una secuencia de imágenes de diferentes vistas del objeto. Para una correcta adquisición, el objeto ha de permanecer estático sin ninguna parte móvil. Este método utiliza algoritmos similares a los utilizados en estereoscopia y es muy sensible al ruido de la imagen de video. La idea básica del sistema es la identificación de puntos comunes en la secuencia de imágenes y el registro de esos puntos en una escena virtual tridimensional. En ocasiones se producen errores debido a que no hay un conocimiento previo de la posición de la cámara y el objeto. Como ventajas de esta técnica hay que resaltar el bajo coste, lo transportable que es el equipo y la capacidad de capturar textura y geometría. La desventaja principal es la baja resolución en la captura de la geometría del objeto.

Adquisición mediante la detección de la silueta Esta técnica se basa en la captura de una sucesión de fotografías alrededor de un objeto, desde diferentes ángulos, contra un fondo que ofrezca un gran contraste con respecto a éste. A partir de esta serie de fotografías se calcula la geometría usando las siluetas extraídas de cada una de ellas (figura 2.3). En algunos sistemas desarrollados recientemente se ha añadido el uso de la información de textura para la corrección y mejora de los datos geométricos calculados. Esta técnica ofrece una alta productividad y un relativo bajo coste. Otra ventajas que ofrecen son la transportabilidad y facilidad de uso del sistema. La principales desventajas son la baja-media resolución que alcanza y las limitaciones para capturar las concavidades de algunos objetos.

Adquisición mediante textura La textura de los objetos puede ser una fuente de información significativa sobre la geometría de una superficie. Se sabe que el sistema visual del hombre puede identificar fácilmente la geometría de una superficie cuando la textura es homogénea. La idea de este método es la identificación de zonas texturadas y encontrar la posible transformación para determinar la variación de la superficie. Es una técnica fotográfica, de bajo coste pero con baja precisión y aplicaciones limitadas (captura de tejidos, piel humana, etc).

Digitalización mediante sombreado Las sombras son una fuente de información en la percepción de la profundidad. Muchos investigadores han intentando simular la manera en que el sistema de visión humana utiliza la información de las sombras para percibir la

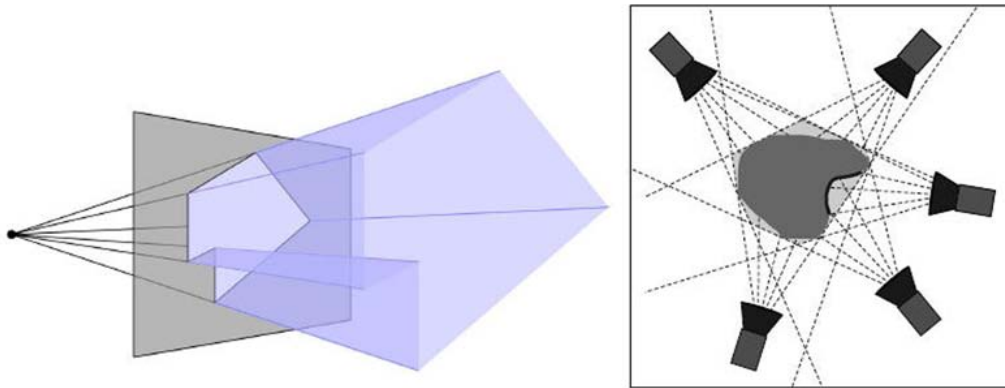


Figura 2.3: Izquierda: Cono de visión limitado por la silueta de la imagen y que contiene al objeto. Derecha: Intersección de varios conos de visión.

profundidad. Este método realiza la captura del objeto desde un punto de vista fijo. Lo que se varía es la posición de la fuente de luz lo que da lugar a una variación de la sombra sobre la superficie del objeto. Así, los algoritmos obtienen la geometría de la superficie del objeto mediante varias fotos con diferentes condiciones de iluminación. Es un método muy simple, con un muy bajo coste y que puede capturar la textura y geometría del objeto. Las principales desventajas son que no puede capturar la textura en las zonas con sombra y que tienen muy baja precisión. Hay una variante de este método, con mejoras en la precisión, que combina esta idea con la del estereoscopio.

Adquisición mediante fotometría Esta es una variante de la adquisición mediante sombreado. En este método se capturan fotos del objeto desde un punto de vista fijo y se van variando las condiciones de iluminación. En este caso, se utilizan objetos de referencia, tales como la posición de la fuente de luz, para la calibración del sistema. Es un sistema de bajo coste, pero que requiere de unas condiciones de iluminación controladas por lo que se emplazan en laboratorios, eliminando así la característica de la transportabilidad.

Adquisición mediante enfoque En los últimos años se ha estado investigando para calcular la geometría de una escena a partir de la profundidad de campo de una foto. Se trata de una técnica recursiva basada en la toma de fotografías de un objeto mientras se varía continuamente el plano de enfoque. Sabiendo la posición del plano de enfoque (a través del posicionamiento del sistema) puede determinarse la profundidad de los píxeles enfocados. De forma recursiva, el sistema va reconstruyendo la geometría del objeto foto a foto. Es una técnica que ofrece una resolución y precisión limitadas. Tiene especial aplicación en combinación con imágenes microscópicas. El coste del sistema es relativamente alto, aunque es muy simple y fácil de aplicar.

Técnicas de visión activa

Estas técnicas hacen intervenir una fuente de luz específica para explorar el entorno. Mediante un emisor y teniendo en cuenta las características de la información recibida, se determina la profundidad de la escena.

A continuación se exponen algunos sensores que utilizan técnicas de visión activa.:

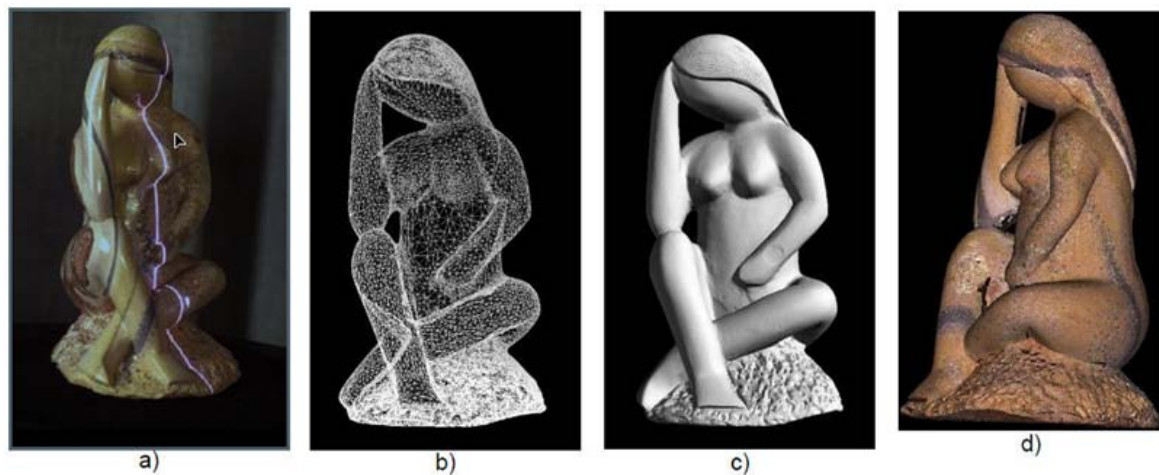


Figura 2.4: Digitalización mediante escaner láser: (a) Barrido del láser sobre el objeto; (b) malla poligonal del objeto digitalizado; (c) representación suavizada de la malla poligonal; (d) inclusión de la textura en la representación del objeto virtual.

Escanners láser Los sistemas de escaneado mediante láser contienen básicamente un emisor láser y un detector óptico. El emisor láser emite un haz con forma de línea o un patrón determinado sobre la superficie de los objetos y el detector óptico (normalmente una cámara digital) detecta esa línea o patrón en los objetos. Aplicando el principio de triangulación extrae la geometría de los objetos (figura 2.4). La ventaja de utilizar emisores láser es que puede adquirir superficies a largas distancias. Además, estos escanners tienen una gran precisión en la medida de características geométricas del objeto. Como desventajas citar la dependencia con las características de la superficie como la reflectancia o transparencia, así como el alto coste de estos equipos. Respecto a la productividad y la transportabilidad, éstas varían significativamente en función del sistema utilizado.

Escanners de luz estructurada Este método se basa en la proyección de uno o varios patrones específicos sobre la superficie de los objetos. A partir de las deformaciones de estos patrones se extrae la información geométrica de los objetos, tras haber aplicado el principio de triangulación. Los patrones pueden ser de diversas formas tales como simples franjas multicolores o complejos patrones con curvas que pueden estar codificados en función del tiempo (figura 2.5). El método también puede ser acompañado de una adquisición de textura y da lugar a resultados de gran realismo y exactitud. Una de las ventajas de estos sistemas es la velocidad ya que en lugar de adquirir un punto están adquiriendo múltiples puntos simultáneamente. Por otra parte, hay que destacar como ventaja que determinados sistemas pueden utilizarse en movimiento. Puede decirse que, actualmente, la investigación en estos sistemas es un área muy activa.

Adquisición mediante sombras Esta técnica es una variante de la adquisición mediante proyección de luz estructurada, aunque en este caso se produce una sombra mediante un objeto simple móvil, o con una fuente de luz móvil. A través de la deformación de la sombra proyectada sobre el objeto se extrae la información de la geometría de la superficie. La principal ventaja de este método es el bajo coste pero, como es evidente, ofrece una baja precisión.



Figura 2.5: Digitalización mediante la proyección de luz estructurada multicolor.

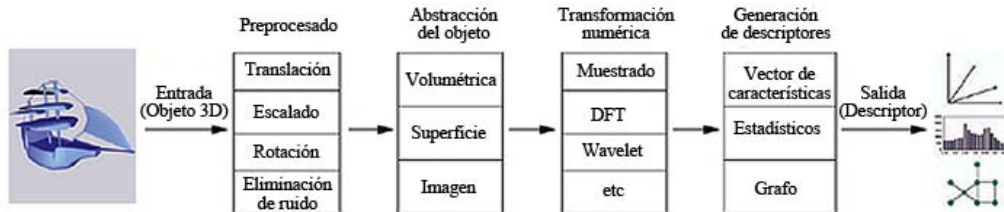


Figura 2.6: Proceso de extracción de características en un objeto 3D.

2.2. Registro

Cuando se digitaliza un objeto, se obtienen varias vistas parciales del mismo, cada una de ellas con su sistema de coordenadas propio. En la fase de registro se calculan las transformaciones a aplicar a cada una de esas vistas parciales para obtener la representación de todas en un sistema de referencia común, y se procede a aplicarlas a cada malla. Por lo tanto el problema principal será obtener las matrices de rotación y traslación con suficiente precisión para que el acople de unas vistas con otras, en el sistema de referencia común, se realice con un error mínimo. Esta etapa de registro es de gran importancia en los procesos de segmentación y reconocimiento de objetos.

Para llevarla a cabo pueden darse las siguientes posibilidades:

1. No existe una estimación de las matrices de transformación.
2. Existe una estimación de las matrices de transformación.

Para el primer caso se utiliza alguna característica de las superficies adquiridas para efectuar la alineación. Estas características pueden ser primitivas básicas en que pueda descomponerse el objeto, descriptores numéricos de ese conjunto de datos que representa el objeto 3D, o cualquier otro descriptor que se defina y que pueda dar información suficiente para efectuar una alineación con precisión. En la figura 2.6 puede observarse un esquema del proceso de extracción de características en un objeto 3D.

La situación más habitual es que exista una estimación de las matrices de transformación ya que los dispositivos con que suelen realizarse las adquisiciones posicionan el objeto frente al sensor, con lo que se conoce el movimiento que ha realizado. Un método muy extendido para el registro o alineación de dos nubes de puntos es el llamado Iterative Closest Point (ICP) [Bes92]. Este método calcula la transformación partiendo de dos vistas parciales en las que existe, como se ha dicho, una estimación de dicha transformación. Partiendo de un conjunto de datos de rango P y un subconjunto de éste Q , se calcula para cada punto $p \in P$ el punto $q \in Q$ más cercano a él. Una vez realizado este paso se evalúa la transformación T (rotación más traslación) que minimiza la suma S de los cuadrados de las distancias entre los

pares (p, q) . Por último se aplica T sobre P y se repite el algoritmo hasta que S llegue a un mínimo. Para que se obtengan buenos resultados es necesario que el error en la estimación inicial de la transformación sea pequeño para que se llegue a una convergencia aceptable. A partir de esta idea básica han sido desarrolladas multitud de variantes de esta técnica.

2.3. Representación de los datos e integración

En esta etapa se lleva a cabo la definición matemática de la superficie que mejor se aproxime al conjunto de vistas parciales, convenientemente registradas. Existen muchos métodos que se pueden emplear para modelar los datos muestreados de una superficie. A continuación se muestra una clasificación de esos métodos, agrupados según el nivel en el que se está trabajando:

- Datos en bruto:
 - Nube de puntos
 - Imagen de rango
 - Sopa de polígonos
- Superficies:
 - Métodos basados en la definición analítica de la superficie
 - Mallas poligonales
- Sólidos:
 - Aproximaciones de partición espacial
 - Arbol BSP
 - Geometría Constructiva de Sólidos (*Constructive Solid Geometry, CSG*)
 - Sweep
- Estructuras de alto nivel:
 - Grafo de la escena
 - Esqueletos (*Skeletons*)

Seguidamente se procederá a describir cada uno de estos métodos, haciéndolo de forma más exhaustiva en los tipos de representación que van a utilizarse en el método de relleno que se propone en la tesis: imagen de rango (sección 2.3.1) y mallas poligonales (sección 2.3.2).

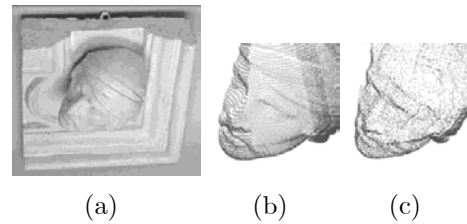


Figura 2.7: a) Fotografía de la escultura que va a digitalizarse; b) Nube de puntos estructurada; c) Nube de puntos desorganizada.

2.3.1. Datos en bruto

Nube de puntos

Se trata de un conjunto de puntos definidos por sus coordenadas, usualmente cartesianas X , Y y Z en un sistema de referencia determinado. Es el tipo de datos que producen los escaners 3D tras el muestreo del entorno. A la hora de generar los datos, pueden darse dos formas distintas de almacenarlos (figura 2.7):

- Nube de puntos desorganizada: Es una mera lista de todos los vértices que componen la nube que no contiene ninguna información topológica. Este tipo de datos suele ser generado por los Sistemas de medida de coordenadas de piezas.
- Nube de puntos estructurada: Es una nube de puntos con un orden determinado, usualmente dispuestos en una matriz que se genera en función de cómo hace el muestro el digitalizador, por lo que el orden en que están dispuestos incluye información topológica. Es la forma más común de generación de datos de los escáneres 3D.

Las nubes de puntos son de utilidad en metrología e inspección de piezas, en donde los puntos muestreados son comparados con el diseño CAD de la pieza para detectar imperfecciones en el proceso de fabricación. Otra de sus aplicaciones es la representación de información volumétrica en imágenes médicas. Por último cabe destacar su aplicación en sistemas de información geométrica, en donde representan el punto de partida para realizar la elevación del terreno.

Como es evidente, las nubes de puntos no ofrecen una representación de la superficie de los objetos, para el estudio de la misma (curvatura, colisiones, etc) o para una visualización más realista (p. ej. con aplicación de textura). Por lo tanto, lo más usual es que sean convertidas a otro tipo de representación que sí la ofrezcan, como las que serán explicadas cuando se trate el tipo de representación: superficies.

Imagen de rango

La imagen de rango es una colección de medidas de distancia a los puntos de la superficie de un objeto de una escena, desde un sistema de referencia conocido. Las imágenes de rango son denominadas de diferentes formas, según el contexto de trabajo: mapa de rango, imagen 3D, imagen 2.5D, mapa de terreno digital (DTM), mapa topográfico, perfil de superficie, lista de puntos xyz , mapa de contorno, mapa de alturas de superficie.

Las imágenes de rango pueden ser representadas de dos formas básicamente. Una es mediante una lista de coordenadas 3D en un sistema de referencia dado, en la que no se

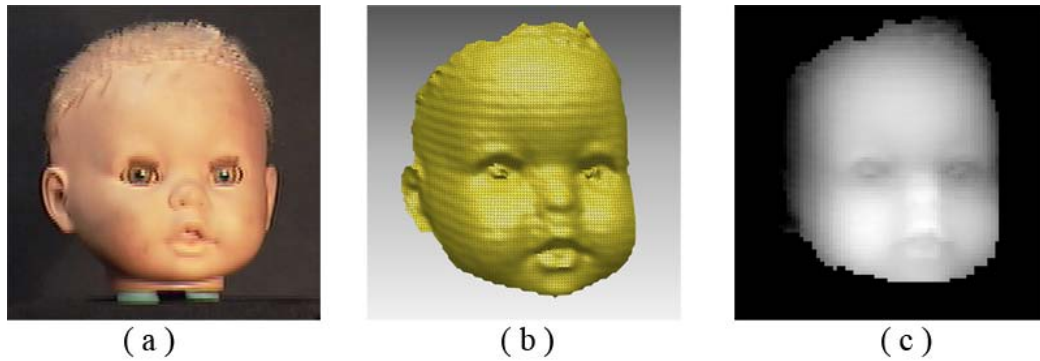


Figura 2.8: Diferentes representaciones de un objeto: (a) fotografía para observar la información del color. (b) malla poligonal obtenida correspondiente a la digitalización de una vista parcial del objeto. (c) imagen de rango o niveles de profundidad correspondientes a esa vista parcial digitalizada.

requiere ningún orden específico. En este caso se dice que la imagen de rango está en forma xyz . La otra forma es una matriz $R(i, j)$ de valores de profundidad a lo largo de las direcciones de los ejes de la imagen x e y , lo que conlleva una organización espacial explícita. Esta forma de representación se denomina forma r_{ij} .

Si los intervalos de muestreo son consistentes en las direcciones x e y de una imagen de rango xyz , podría representarse mediante una matriz de valores r_{ij} de rango cuantizados y escalados, donde cada coordenada xyz vienen determinadas de forma implícita por la fila y columna que ocupa en la matriz y el valor de rango almacenado en la celda. El término “imagen” se utiliza porque cualquier imagen de rango r_{ij} puede ser visualizada como una imagen en niveles de gris, en la que cada uno de esos niveles corresponde a diferentes valores de profundidad. Si r_{ij} es el valor de un pixel en la i -ésima fila y j -ésima columna de la matriz, las coordenadas 3D correspondientes vendrán dadas por:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_x + s_x i \\ a_y + s_y j \\ a_z + r_{ij} i \end{pmatrix} \quad (2.1)$$

en donde los valores s_x, s_y, s_z son los factores de escala y los valores a_x, a_y, a_z son los offsets. En la figura 2.8 aparecen diferentes representaciones de la vista parcial de un objeto digitalizado: en la figura 2.8-a se observa la fotografía correspondiente a esa vista parcial en la que se muestra la información de color; en la figura 2.8-b puede verse la malla poligonal correspondiente a esa vista; en la figura 2.8-c se ha representado la imagen de rango de esa digitalización; en la figura 2.9 se representa en forma de mapa de alturas la información contenida en la imagen de rango de la figura 2.8. Cada una de estas representaciones es igualmente válida para visualizar los datos.

La transformación afín mostrada en la ecuación 2.1 es adecuada para imágenes de rango ortográficas donde las profundidades se miden a lo largo de vectores paralelos, perpendiculares al plano de imagen. En los sensores de rango activos es más común la transformación no afín de los datos de rango r_{ij} a coordenadas cartesianas x, y, z .

En el sistema de coordenadas esféricas mostrados en la figura 2.10, los índices (i, j) se corresponden con los ángulos de latitud (elevación) y longitud (azimuth) respectivamente. La transformación de coordenadas esféricas a cartesianas se haría como sigue:

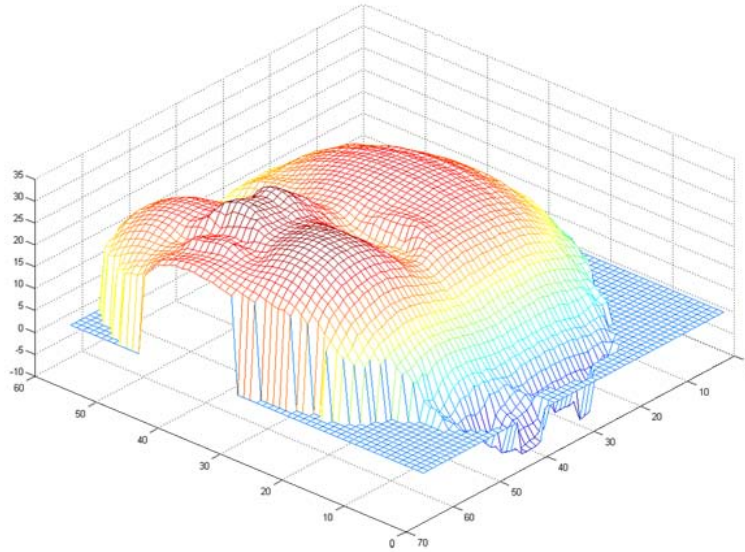


Figura 2.9: Representación del mapa de alturas correspondiente a una imagen de rango.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_x + s_r r_{ij} \cos(is_\phi) \text{sen}(js_\theta) \\ a_y + s_r r_{ij} \text{sen}(is_\phi) \\ a_z + s_r r_{ij} \cos(is_\phi) \cos(js_\theta) \end{pmatrix} \quad (2.2)$$

donde s_r , s_ϕ , s_θ son los factores de escala en rango, latitud, longitud y los valores a_x , a_y , a_z son los offsets. El sistema de coordenadas con eje ortogonal alternativo, también mostrado en la figura 2.10, utiliza otro ángulo de elevación ψ . En este caso la conversión a coordenadas cartesianas sería:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_x + \frac{s_r r_{ij} \tan(js_\theta)}{\sqrt{1+\tan^2(is_\theta)+\tan^2(js_\psi)}} \\ a_y + \frac{s_r r_{ij} \tan(js_\psi)}{\sqrt{1+\tan^2(is_\theta)+\tan^2(js_\psi)}} \\ a_z + \frac{s_r r_{ij} \tan(js_\psi)}{\sqrt{1+\tan^2(is_\theta)+\tan^2(js_\psi)}} \end{pmatrix} \quad (2.3)$$

el nuevo ángulo de elevación ψ depende solamente de y y z mientras que ϕ depende de x , y y z . Las diferencias en (x, y, z) para las ecuaciones 2.2 y 2.3 para el mismo valor de azimuth y elevación son menores del 4% en x y z y menos del 11% en y , incluso cuando ambos ángulos son del orden de 30° .

Sopa de polígonos

En esta representación se tiene es una colección desordenada de polígonos sin información de las posiciones relativas de unos polígonos respecto de otros. Se almacena solamente una lista de caras, especificando la posición de los vértices de cada cara, por lo que no incluye relaciones topológicas entre las mismas Podemos considerar que los algoritmos que utilizan esta representación suelen ser menos eficientes y robustos que aquellos que poseen información topológica adicional. A pesar de esta desventaja resulta más fácil para algunos usuarios y creadores realizar un diseño mediante esta estructura ya que no les obliga a tener en

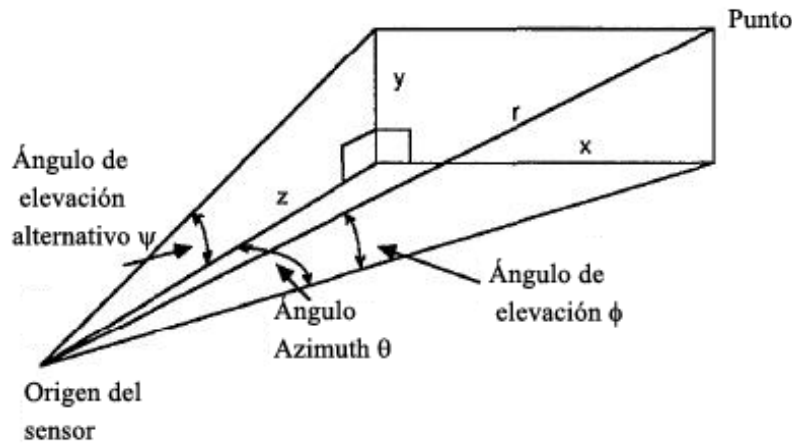


Figura 2.10: Coordenadas cartesianas, esféricas y ejes ortogonales.

mente una determinada estructura. Es también conocida como *trimesh* en algunos motores de cálculo de físicas (colisiones, dinámicas,...).

En esta estructura se utilizan dos matrices para almacenar los datos: una matriz V que contiene todas las coordenadas de todos los vértices de todas las caras y una matriz F de índices a vértices, con una entrada para cada cara. De esta forma, se dispone de una estructura estática, que no puede ser procesada lo que dificulta su análisis, además de ser algo ineficiente en el sentido de que los vértices pueden estar repetidos muchas veces. Como se verá, a continuación, la estructura resulta parecida a las mallas poligonales, sin embargo, en este caso, las caras son de forma y tamaño variable, además de que no existe ninguna relación entre ellas, y, de hecho, puede darse que caras adyacentes tengan diferente número de vértices en la arista por la que están unidas.

Como ventajas cabe destacar que ofrecen una gran facilidad en la representación de discontinuidades de profundidad sin ninguna información extra, y la continuidad de la superficie 3D puede ser asegurada al fijar las mismas coordenadas en determinados vértices de polígonos distintos.

Estas estructuras son muy utilizadas en el ámbito de la programación en 3D en tiempo real, por ejemplo en programación de videojuegos. En la figura 2.11 aparecen diferentes ejemplos de la representación tipo sopa de polígonos.

2.3.2. Superficies

Métodos basados en la definición analítica de la superficie

Este tipo de representaciones son completas, de forma que la descripción geométrica es explícita, la totalidad de la superficie o del objeto es descrita, y por tanto, la representación del objeto puede hacerse para cualquier posición arbitraria del mismo.

Funciones paramétricas La forma paramétrica genérica de una superficie 3D es [Cam01]:

$$S(u, v) = \begin{bmatrix} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{bmatrix} \quad (2.4)$$

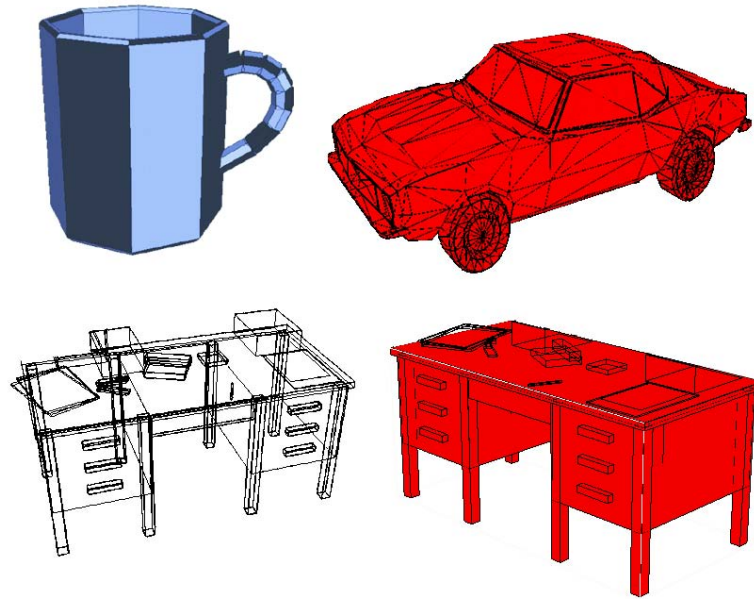


Figura 2.11: Diferentes ejemplos de representación utilizando el método sopa de polígonos.

Las tres funciones $f(u, v)$, $g(u, v)$ y $h(u, v)$ tienen como argumentos las dos variables paramétricas (u, v) , por lo que se dice que es una superficie biparamétrica (figura 2.12). Sin pérdida de generalidad, el dominio de (u, v) puede ser restringido a un cuadrado unitario $[0, 1] \times [0, 1]$. Si uno de los valores de los parámetros se mantiene constante mientras el otro varía, se genera una curva isoparamétrica en la superficie, que también puede observarse en la figura 2.12. Si, en cambio, se mantienen constantes ambos parámetros se genera un punto. Los límites de la superficie se obtienen manteniendo el máximo o mínimo valor de uno de los parámetros y variando el otro. Si se define uno de los parámetros como una función del otro también se obtienen curvas en la superficie. Por ejemplo para una superficie con límites $0 \leq u \leq 1$ y $0 \leq v \leq 1$, las funciones $u = 1 - v$ y $v = 1 - u$ generan las diagonales de la superficie.

De entre las técnicas existentes para la generación de superficies mediante funciones paramétricas, la más común es la conocida como NURBS (Nonuniform rational B-Spline) propuesta por en [Ver75]. Un superficie NURBS $S(u, v)$ de grado (p, q) es una función racional bivariada de la forma [Igl02]:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m P_{ij} \omega_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^n \sum_{j=0}^m \omega_{i,j} N_{i,p}(u) N_{j,q}(v)} \quad (2.5)$$

donde $\{\omega_{ij}\}_{i,j}$ representan los valores escalares de los pesos asociados a los puntos de control $\{P_{ij}\}_{i=0,\dots,n;j=0,\dots,m}$ y $N_{i,p}(u)$ y $N_{j,q}(v)$ son las funciones bases normalizadas B-Splines de

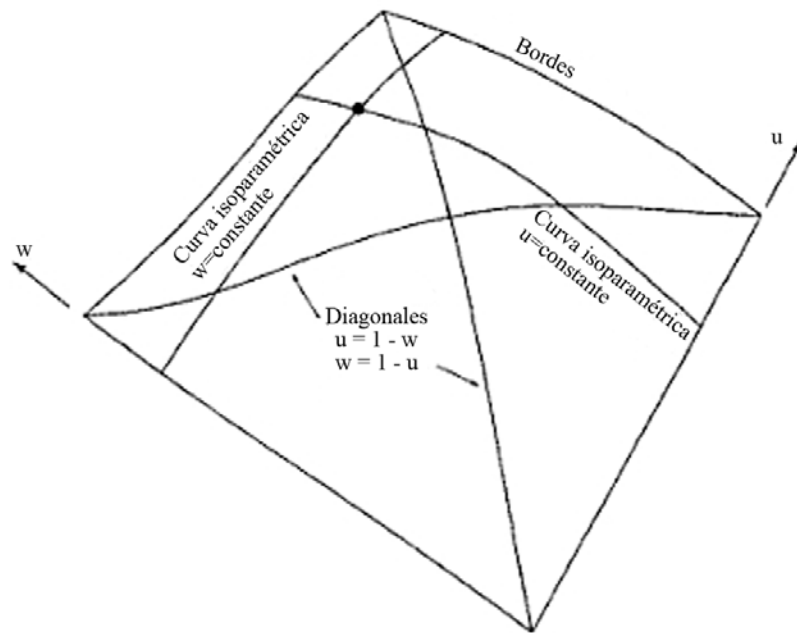


Figura 2.12: Representación de una superficie paramétrica, con algunas curvas contenidas en la misma.[Rog01]

grado p y q , respectivamente. Una función básica i -ésima B-Spline de grado p (orden $p + 1$), denotada por N , está definida recursivamente como sigue:

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1} \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (2.6)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+1} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (2.7)$$

donde u_i es el vector de nudos (*knot*). Este último vector indica los puntos en donde la derivada de la superficie puede tener una discontinuidad.

Las superficies NURBS son funciones racionales definidas por intervalos de forma paramétrica. Estas son expresadas en terminos de una red de control, en la que cada punto actúa como un atractor para la superficie resultante. En la figura 2.13 se muestra un ejemplo en el que las líneas sobre la superficie determinan los cambios en la definición de las NURBS (*knot lines*).

Las superficies NURBS son útiles por diversas razones:

- Son invariantes a las transformaciones afines y de perspectiva: operaciones como rotaciones y traslaciones pueden ser aplicadas a las NURBS solo con su aplicación a la red de puntos de control.
- Ofrecen flexibilidad para el diseño interactivo, con una gran habilidad para representar de manera precisa formas cerradas, cónicas y cuádricas.
- Necesitan menos espacio en memoria para ser almacenados que otros métodos que requieren que se almacenen todas las entidades en que dividen a la superficie, como por ejemplo las sopas de polígonos o las mallas poligonales..

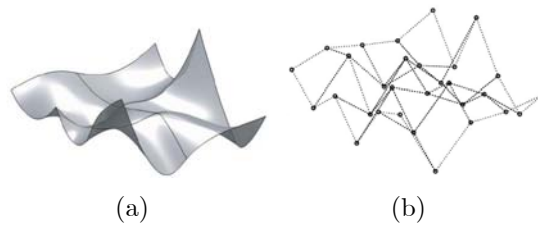


Figura 2.13: (a) Superficie NURB con líneas de nudos (knot lines); (b) Red de puntos de control [Sev09].

El principal inconveniente de este método es el gran coste computacional que implica su cálculo. Por otro lado P.J. Besl en [Bes90] indica que este tipo de representaciones no son muy adecuadas para tareas de visión por computador debido, entre otras cosas, a que los puntos de control no son fácilmente detectables ni únicos. Para profundizar en este tema puede recurrirse al libro de Rogers [Rog01].

Funcion implícita Una función implícita es una función multivariable definida en un dominio \mathbb{R}^n . En el espacio 3D, viene dada por una función

$$F : \mathbb{R}^3 \rightarrow \mathbb{R} \quad (2.8)$$

Tal que, para un isovalor v , la superficie implícita, o isosuperficie, se define por $F(x, y, z) = v$ o:

$$F(x, y, z) - v = 0 \quad (2.9)$$

En la representación de superficies de objetos, estas vienen representadas como el conjunto de datos en los que una función, f , se hace igual a cero:

$$S = \{(x, y, z) / f(x, y, z) = 0\} \quad (2.10)$$

De esta forma, la función implícita se define de forma que sea negativa dentro de la superficie, nula en la superficie, y positiva fuera de la misma.

Se han utilizado diversos tipos de funciones para describir superficies implícitas. Las funciones cuadráticas pueden utilizarse para describir superficies naturales tales como las esferas, elipsoides y cilindros [Lev76]. Las funciones polinómicas de un grado mayor pueden utilizarse para crear formas más complejas, aunque se hace difícil el control de estas formas [Ker04], [Tau96]. Los parches algébricos pueden utilizarse para mejorar el control en la creación de la superficie, aunque resulta costoso en términos de computación conseguir una unión suave entre los parches [Baj95]. Estos tipos de superficies son descritas con gran detalle en el libro editado por Bloomenthal [Blo97].

Como ejemplo se muestra en la figura 2.14(a) la superficie implícita de un toro, cuya ecuación es la siguiente:

$$f(P) = x^4 + y^4 + z^4 + R^4 + r^4 + 2(x^2y^2 + x^2z^2 + y^2z^2 - (R^2 + r^2)(x^2 + z^2) + (R^2 - r^2)y^2 - R^2r^2) \quad (2.11)$$

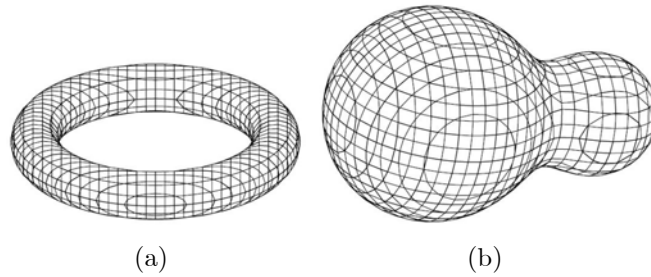


Figura 2.14: (a) Representación de la superficie implícita de un toro; (b) representación de la superficie "blobby" formada por la unión de dos esferas.

Con respecto a las superficies paramétricas presenta algunas propiedades ventajosas. Por ejemplo ofrece un fácil acceso a la evaluación de la superficie, buenas propiedades para el modelado, buenas características de extrapolación de datos y posee la capacidad de poder representar objetos de cualquier topología.

La reconstrucción de superficies implícitas o lo que es lo mismo la generación de la función implícita que define una superficie a partir de los datos de rango puede hacerse usando diversos métodos. La mayoría de ellos están basados en la idea de Blinn de unir o fusionar primitivas implícitas [Bli82]. Los "blobs", como él llamó a las superficies implícitas primitivas, dan lugar a un efecto de unión de *metaballs* que ha sido usado, por ejemplo, en el diseño de formas orgánicas. En la figura 2.14(b) se muestra un ejemplo de este tipo de superficies "blobby" cuya ecuación de superficie es:

$$f(P) = g(P) + h(P) \quad (2.12)$$

donde $g(P)$ y $h(P)$ son las dos esferas cuya unión componen la superficie, y que están definidas por las siguientes ecuaciones:

$$g(P) = \frac{r_g^2}{|Centro_g - P|^2} \quad (2.13)$$

$$h(P) = \frac{r_h^2}{|Centro_h - P|^2}$$

siendo r_g y r_h los radios de ambas esferas y c_g y c_h las coordenadas de sus respectivos centros.

La desventaja que poseen es en la visualización de las mismas, y en la creación de algunas formas complejas. En este sentido suele interesar la conversión a modelos poligonales, aunque esta debe realizarse cada vez que se cambie la superficie implícita original (figura 2.15).

Mallas poligonales

Una malla poligonal se podría definir como el par

$$\mathcal{O} = \langle \mathcal{P}, \mathcal{V} \rangle \quad (2.14)$$

donde $\mathcal{V} = \{v_1, \dots, v_n\}$ es el conjunto de vértices o nodos 3D que se aproximan a los datos de rango, siendo $v_i = (x_i, y_i, z_i)^T$, y $\mathcal{P} = \{p_1, \dots, p_m\}$ una lista de elementos que relacionan a

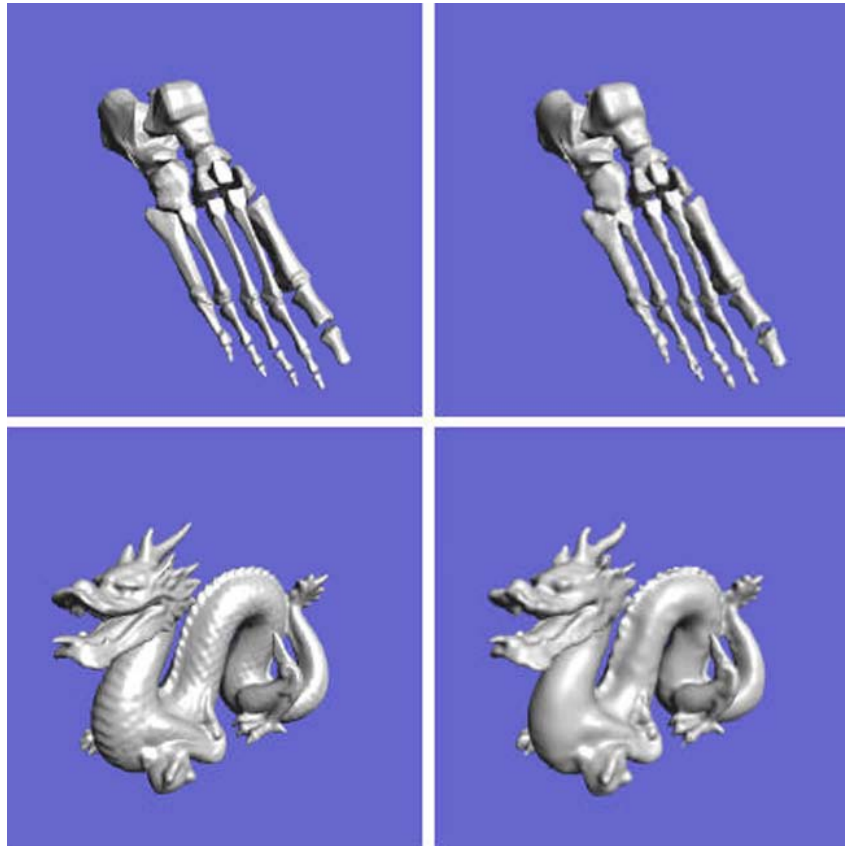


Figura 2.15: Conversión de malla poligonal a superficie implícita [Tur01].

los vértices entre sí. Estas relaciones suelen ser las aristas que unen a un par de vértices y/o los polígonos o patches por formados por éstos.

Se pone de manifiesto, tras esta formalización, que las mallas poligonales vendrán caracterizadas fundamentalmente por dos propiedades:

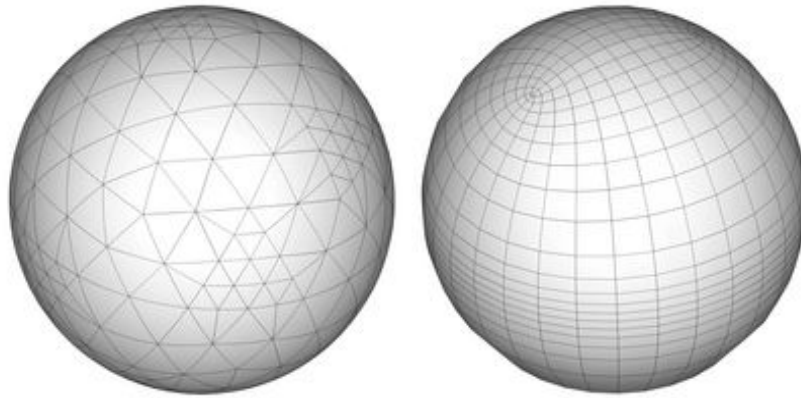
- **Propiedades geométricas** que están relacionadas con las coordenadas de los vértices, o lo que es lo mismo, con la posición de éstos en el espacio 3D.
- **Propiedades topológicas** referidas a la relación existente entre los vértices o nodos de la malla.

Centrándonos en las propiedades topológicas, se puede realizar una clasificación de las mallas en función de la información que almacene \mathcal{P} .

Si en $\mathcal{P} = \{p_1, \dots, p_m\}$ se almacenan las aristas que unen a los distintos vértices, se podrá calcular la relación de vecindad entre ellos. Teniendo en cuenta esta relación, la malla se puede clasificar como:

- **Malla no estructurada** en donde el número de vecinos de cada nodo no es constante.
- **Malla estructurada** en donde el número de vecinos de cada nodo es constante.

Las segundas presentan ciertas ventajas sobre las primeras ya que son más simples, requieren menos memoria para su almacenamiento en un ordenador y permiten un control más directo sobre la forma de los patches. Además, otra de sus ventajas es la existencia de multitud de algoritmos más eficientes en tiempo de computación para las mallas estructuradas.



Structured (orthogonal tripolar) vs. unstructured (icosahedral quadtree) spherical grids.

Figura 2.16: Representación de una esfera mediante: (a) Malla estructurada; (b) Malla no estructurada.

Por otro lado, al ser el número de vecinos un parámetro fijo, es más sencillo el cálculo de propiedades sobre los nodos. La gran desventaja es la poca flexibilidad que presentan a la hora de aproximarse a superficies complejas, presentándose en las mallas obtenidas un suavizado en las zonas de mayor curvatura de los objetos que se pretendan modelar.

La decisión sobre la utilización de un tipo u otro de malla dependerá de la aplicación a la que vayan destinados los modelos. De esta forma, las mallas estructuradas no serán aceptables cuando se necesite una representación exacta del objeto, como puede ser en una aplicación CAD. En la figura 2.16 puede observarse la representación de una esfera mediante los dos tipos de malla mencionados.

Otra clasificación que se puede realizar basándonos en las propiedades topológicas de la malla es en función del tipo de polígono o patch que se utilice en éstas. En este caso, cada elemento $t_i \in \mathcal{T}$ almacenará una lista con los vértices que definen a los polígonos de la malla. En principio el tipo de patch puede ser de muy distinto tipo, pero los más habituales son:

- **Triangulares**
- **Cuadrangulares**
- **Hexagonales**

Las mallas triangulares son las más más utilizadas por dos razones: es el tipo de polígono más simple y permiten calcular los vectores normales a las superficies necesarios para la simulación del proceso de reflexión de la luz sobre éstas. Esta segunda propiedad es empleada en aplicaciones relacionadas con gráficas por computador. Cuando se utilizan este tipo de patches, las mallas suelen ser no estructuradas.

Las mallas con patches cuadrados están relacionadas esencialmente con la obtención de éstas a partir de las representaciones obtenidas mediante técnicas analíticas, sobre todo con las funciones paramétricas. No suelen ser empleadas cuando se trabaja en tareas de visión por computador.

Por último, las mallas hexagonales son mallas estructuradas con tres vecinos por nodo. Una característica interesante de este tipo de mallas es que su dual es una malla triangular (figura 2.17).

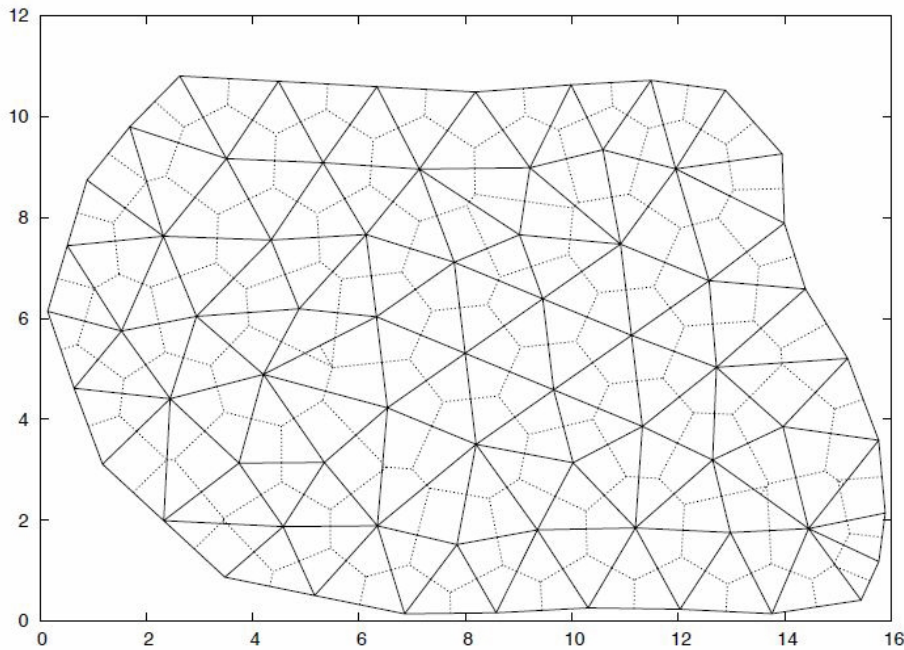


Figura 2.17: Representación de una malla hexagonal en 2D y su dual triangular [Mar07].

El tipo de mallas que se utilizarán en los algoritmos planteados en esta tesis serán las mallas triangulares, que serán explicadas a continuación.

Mallas triangulares.

Una malla triangular \mathcal{M} consiste en un componente geométrico y otro topológico en el que la superficie viene representada por un conjunto de vértices

$$\mathcal{V} = \{v_1, \dots, v_V\} \quad (2.15)$$

y un conjunto de caras triangulares que los conectan

$$\mathcal{T} = \{t_1, \dots, t_m\}, \quad t_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V} \quad (2.16)$$

en donde cada triángulo especifica los vértices de que lo forman. La conectividad de la malla también viene definida en términos de las aristas

$$\mathcal{A} = \{a_1, \dots, a_k\}, \quad a_i \in \mathcal{V} \times \mathcal{V} \quad (2.17)$$

La componente geométrica de la malla triangular en \mathbb{R}^3 viene especificada mediante la asociación de la posición 3D p_i a cada vértice $v_i \in \mathcal{V}$:

$$\mathcal{P} = \{p_1, \dots, p_V\}, \quad p_i := p(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix} \in \mathbb{R}^3, \quad (2.18)$$

La superficie está formada, por tanto, por una red de triángulos cuyos vértices son los puntos de la nube \mathcal{P} y cuya conectividad viene definida por la 3-tupla de elementos $\{\mathcal{V}, \mathcal{A}, \mathcal{T}\}$. Se puede decir que la malla triangular es un grafo $G(V, A)$ en el que se establece una relación de conectividad entre los vértices mediante las aristas, tal y como puede observarse en la

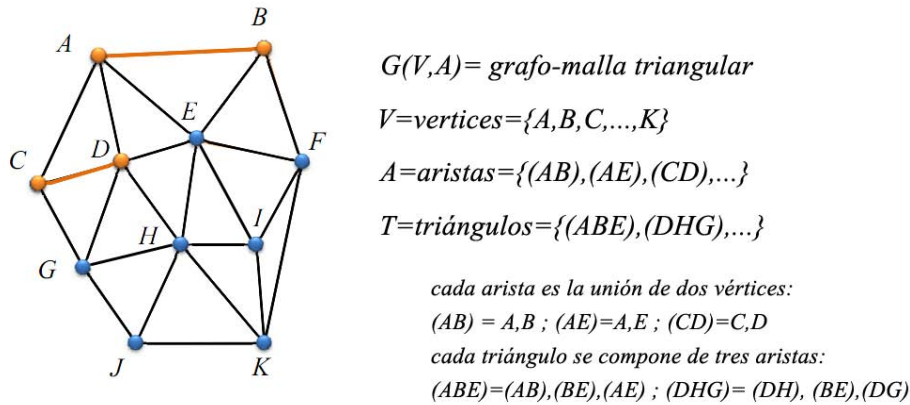


Figura 2.18: Entidades que componen la malla triangular e interconexión entre ellas

figura 2.18. Los triángulos están formados por la conexión de tres vértices vecinos en el grafo G , mediante las respectivas tres aristas.

Una mallas triangular representa por tanto una superficie lineal continua definida a tramos. Si se utiliza para aproximar una superficie suficientemente suavizada, se puede determinar mediante una expansión local de Taylor que el error es del orden de $O(h^2)$, siendo h la longitud máxima de aristas. Debido a esta aproximación cuadrática, el error puede reducirse en $1/4$ si se dividen a la mitad la longitud de las aristas. Al realizar esta subdivisión se divide cada triángulo en cuatro subtriángulos, lo que hace que se incremente el número de triángulos de F a $4F$. Por tanto, el error de aproximación de una malla triangular a una superficie es inversamente proporcional al número de caras, y depende del término de mayor exponente de la expansión de Taylor. Debido a esto para conseguir una gran aproximación a la superficie, habrá que adaptar la densidad de vértices a la curvatura de la superficie, de forma que las zonas planas tengan una baja densidad, mientras que en las zonas de detalle la densidad será muy alta.

Una caracterización topológica importante de una superficie es si es o no una variedad bidimensional (en inglés: superficies *2-manifold*), es decir, si no existen puntos singulares donde la superficie se intersece consigo misma o se abra en varias hojas (figura 2.19(a)) se dice que que es una variedad bidimensional, en caso contrario se dice que no lo es (*non-manifold*). Este tipo de mallas son fuente de problemas para la mayoría de los algoritmos, ya que las relaciones de vecindad no quedan bien definidas en el entorno de la configuración sin variedad bidimensional. En la figura 2.19(a) se presentan diversos tipos de topologías de la vecindad de un punto v_i que producen variedades bidimensionales y que no las producen.

Además, el hecho de que una malla triangular sea cerrada implica una continuidad en toda esa red de triángulos que modelan la superficie. Esto se traduce, teniendo en cuenta la forma de construcción de la malla triangular, en que todas las aristas pertenecen únicamente a dos triángulos contiguos en la malla.

De esta forma, se deduce que, para toda malla triangular cerrada y que sea una variedad bidimensional $M = G(V, A)$, se cumplen las siguientes propiedades, referidas cada una de ellas a uno de los elementos de la 3-tupla \mathcal{V} , \mathcal{A} y \mathcal{T} :

i) Conectividad de los vértices: todos los vértices de la malla están conectados con al menos otros tres vértices vecinos, o lo que es lo mismo, tienen al menos tres aristas incidentes: $\forall v_i \in \mathcal{V}, \exists \mathcal{A}_S \subset \mathcal{A} / \forall a_h \in \mathcal{A}_S, v_i \subset a_h \wedge \text{card}(\mathcal{A}_S) \geq 3$

ii) Pertenencia de aristas a triángulos: todas las aristas están compartidas únicamente por dos triángulos, que se dice que son contiguos: $\forall a_h \in \mathcal{A}, \exists ! t_i, t_j / t_i \cap t_j = a_h, i \neq j$

iii) Conexión de los triángulos: cualquier par de triángulos de la red tienen o una arista en común o ninguna; $\forall t_r, t_s \in \mathcal{T}, \text{card}(t_r \cap t_s) \in \{0, 1\}$

Dado que entre los elementos de la 3-tupla existe una relación de conectividad, el enunciado de cada una de esas propiedades viene implícito en las otras propiedades.

Otro concepto que se utiliza para caracterizar a este tipo de mallas es el de grado un vértice, $\text{grad}(v_i)$, que se define como el número de aristas que inciden sobre él. De esta forma, en el caso de mallas triangulares cerradas y que sean una variedad bidimensional siempre se cumplirá que $\text{grad}(v_i) \geq 3, \forall i$. Además si se tratase de una malla totalmente regular, $\forall v_i \in \mathcal{V}, \text{grad}(v_i) = \text{cte}$.

Por otra parte, al tratarse de variedades bidimensionales cerradas, puede utilizarse la fórmula de Euler-Poincaré, para obtener el número de componentes que tiene cada conjunto de entidades, $\mathcal{V}, \mathcal{A}, \mathcal{T}$, que componen la malla. De esta forma, se tiene la siguiente relación:

$$\begin{aligned} \text{card}(\mathcal{V}) + \text{card}(\mathcal{T}) - \text{card}(\mathcal{A}) &= \chi \\ \chi &= 2(1 - g) \end{aligned} \quad (2.19)$$

donde χ es la constante característica de Euler, que será diferente, según las características de la malla; y, $g \in \mathbb{N}$ es el género de la superficie y representa intuitivamente el número de asas de un objeto. Dado que lo más habitual en todas las mallas es que el género sea pequeño en comparación con el número de elementos, la constante χ puede aproximarse a cero para obtener una relación aproximada entre el número de los elementos que la componen. Por otra parte, dado que cada triángulo está limitado por tres aristas, de igual forma que cada arista (interior) es compartida por dos triángulos, pueden obtenerse las siguientes relaciones:

$$3 \cdot \text{card}(\mathcal{T}) \approx 2\text{card}(\mathcal{A}) \quad (2.20)$$

$$\text{card}(\mathcal{A}) \approx 3\text{card}(\mathcal{V}) \quad (2.21)$$

$$\text{card}(\mathcal{T}) \approx 2\text{card}(\mathcal{V}) \quad (2.22)$$

Si se combinan las expresiones (2.19) y (2.20)-(2.22), se deduce que, por término medio, en una malla triangular el grado de todos sus vértices es 6, tal y como se obtiene con las siguientes expresiones:

$$\text{card}(\mathcal{V}) + \text{card}(\mathcal{T}) - \text{card}(\mathcal{A}) = \text{card}(\mathcal{V}) + (2/3)\text{card}(\mathcal{A}) - \text{card}(\mathcal{A}) = 2(1 - g) \quad (2.23)$$

$$\text{card}(\mathcal{A}) = 3(\text{card}(\mathcal{V}) - 2 + 2g) \quad (2.24)$$

A partir de esto y recordando cómo se definía el grado de un vértice, se puede deducir su formulación matemática aproximada, sabiendo que será el cociente entre el número de aristas y la mitad del número de vértices. Así, aplicando la definición y la expresión (2.24):

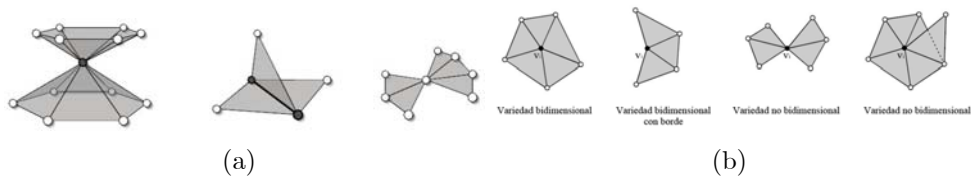


Figura 2.19: (a) Dos partes de la superficie confluyen en un vértice que produce una variedad no bidimensional (a la izquierda). Una arista que no produce una variedad bidimensional tiene más de dos caras incidentes (en el centro). La configuración de la derecha, aunque da lugar también a una variedad no bidimensional, es más habitual en muchas mallas [Bot06]; (b) Vecindad de un vértice dado v_i [Vas06].

$$\forall v_i \in \mathcal{V}, \text{grad}(v_i) \approx 2\text{card}(\mathcal{A})/\text{card}(\mathcal{V}) = 6(\text{card}(\mathcal{V}) - 2 + 2g) / \text{card}(\mathcal{V}) \quad (2.25)$$

Si la expresión se aplica en mallas de objetos, en los cuales normalmente el número de vértices es muy elevado, se tiene que:

$$\text{grad}(v_i) \approx 6$$

En comparación con las spline y las superficies por subdivisión, las mallas triangulares no están restringidas por limitaciones geométricas, topológicas o de conectividad, y pueden por tanto ser consideradas como la representación más flexible de todas. Los triángulos son, conceptualmente, la primitiva más simple para la representación de superficies, y, es por esto que, permiten la implementación de algoritmos muy eficientes.

2.3.3. Sólidos

El modelado de sólidos es una técnica matemática para la representación de objetos sólidos en un ordenador. A diferencia del modelado de superficies, en el modelado de sólidos se asegura una conexión adecuada entre todas las superficies y que el objeto es geoméricamente correcto. El modelado de sólidos añade la complicación de que implica la representación del objeto de forma interna, describiendo el volumen interior, la masa y el peso del objeto. Si se secciona un modelo pueden verse los elementos que lo componen interiormente.

De cualquier forma, a pesar de la complejidad que puede representar un modelo sólido, en la mayoría de las ocasiones suele interesar obtener solamente la representación de la parte visible del mismo, es decir, de la superficie externa.

A continuación se explicarán brevemente las técnicas más utilizadas para la representación de la superficie de los sólidos.

Aproximaciones de partición espacial

Mediante esta representación los sólidos son caracterizados por un conjunto de volúmenes $\{V_i\}$, que permite caracterizar qué parte del espacio euclídeo está dentro del objeto y qué parte está fuera. Esta forma de representación cumple lo siguiente:

- $i) \forall i, V_i \neq \emptyset$
- $ii) \forall i, j, V_i \cap V_j = \emptyset$
- $iii) \cup_i \{V_i\} = \text{Espacio}(\mathbb{R}^3)$

Algunas de las utilidades del empleo de este método de representación son:

- Clasificación de puntos : para discernir cuáles son los puntos del espacio ocupados por el objeto y cuáles no. Esto es
- Información sobre propiedades de puntos en el espacio. Cada volumen de la partición puede además contener información adicional sobre alguna propiedad del objeto en esa parte del espacio (densidad, color, presión, composición. . .).
- Clasificación de objetos dentro de la escena : en este caso no se trataría de describir la forma de un objeto determinado, sino de agrupar los objetos de una escena compleja en partes para optimizar los métodos de visualización.

Por tanto el espacio euclídeo es particionado en regiones tridimensionales no sobrepuestas (pero sí adjuntas) llamadas *celdas* o *vóxeles* [Ros99]. Los distintos esquemas de partición varían según las restricciones que se impongan a las celdas, como forma, posición, orientación, etc. Una clasificación muy utilizada divide a las particiones en jerárquicas y no jerárquicas. En las primeras el sólido se representa mediante en sucesivas particiones de más y más detalle, cuyos volúmenes están incluidos en una partición de nivel inferior, y que se forma mediante operaciones sucesivas de división. Un ejemplo de este tipo de partición, muy utilizado en programación de gráficos, son los árboles octales (*octrees*). En las particiones no jerárquicas, existe un único conjunto de volúmenes que forman la partición, y no tiene relaciones internas de inclusión. Este es el caso de: la descomposición de celdas (*cell decomposition*), la enumeración de la ocupación espacial (*spatial occupancy enumeration*). Pasemos a continuación a analizar estos tipos concretos de representación.

La descomposición de celdas es la forma de particionamiento más simple. que consiste en descomponer el sólido en varias celdas (figura 2.20). Si bien no genera ambigüedad, una representación puede tener varias acepciones. La validación también es costosa, ya que para casi todos los pares de celdas hay que establecer si hay o no intersección. Este método ha sido muy utilizado en el análisis de elementos finitos, así como en la representación del interior de edificios para la planificación de trayectorias con robots móviles [Can88].

La enumeración de la ocupación espacial es un caso particular de la descomposición de celdas donde cada celda en el esquema (llamada *voxel*) debe ser cúbica, del mismo tamaño y recaer en una malla fija [Req80]. La restricción que establece este esquema permite que cada celda sea ubicada con las coordenadas de un solo punto; por lo tanto para representar un objeto solamente es necesario definir una malla 3D y una lista de celdas ocupadas (enumeradas). No se requiere validación ya que la malla y las restricciones de las celdas evitan intersecciones inapropiadas. Se puede hacer una representación del espacio para distintos tamaños de celdas, lo que proporciona la posibilidad de tener información a distintas resoluciones, dependiendo a qué nivel de detalle se quiera trabajar con el sólido, con el consiguiente ahorro de recursos de ordenador (figura 2.21).

Por lo que respecta a los árboles octales u *octrees* es un esquema de representación de objetos en términos de una estructura de árbol octal obtenida por divisiones recursivas

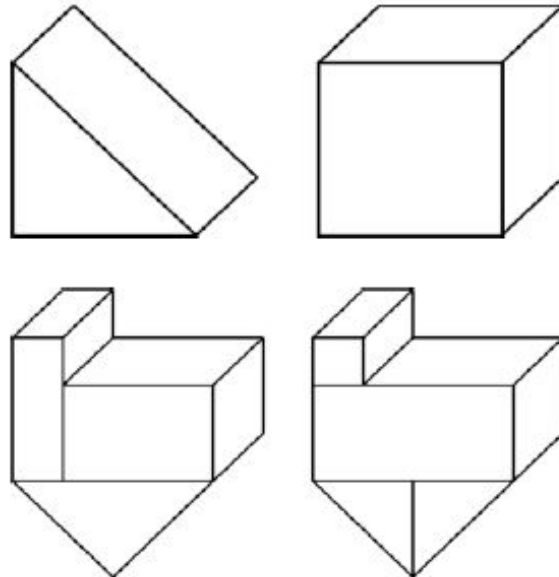


Figura 2.20: Descomposición de un objeto en diferentes celdas básicas.

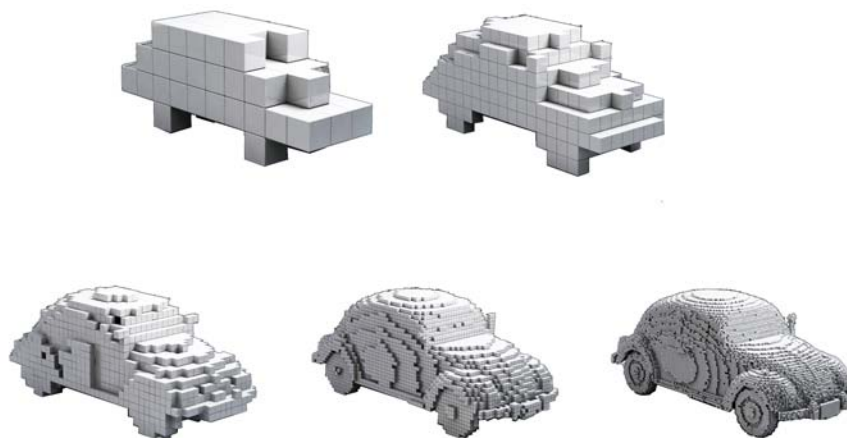


Figura 2.21: Representación mediante vóxeles de distinto tamaño de un mismo objeto.

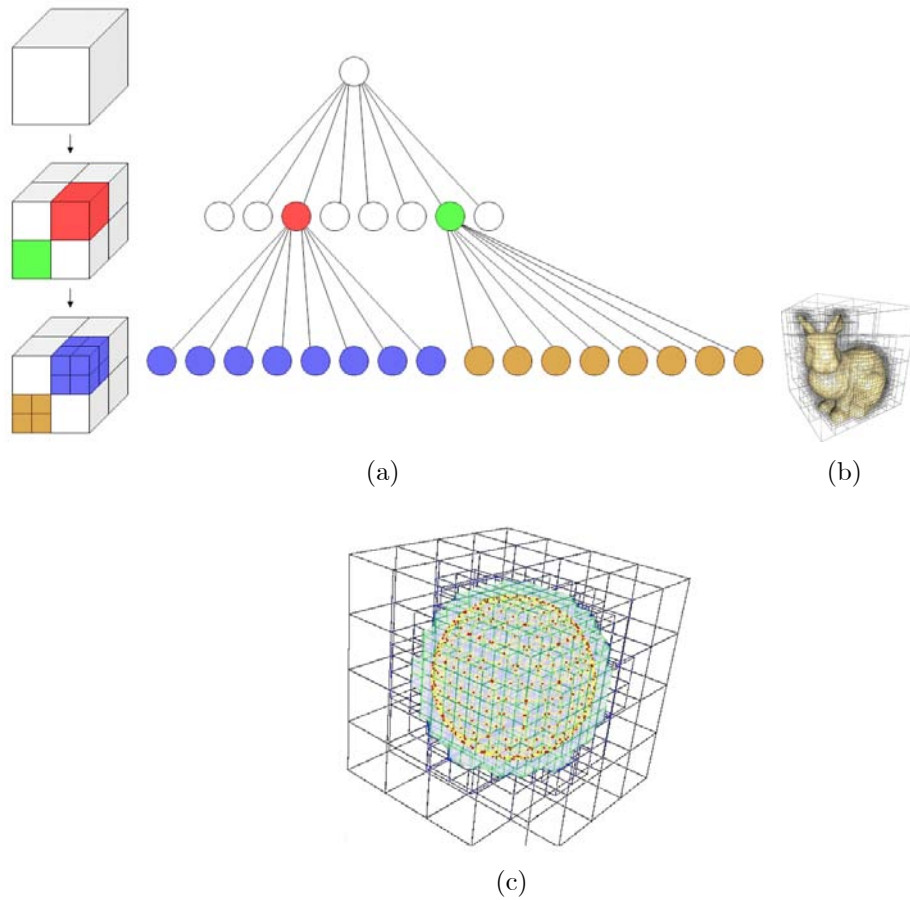


Figura 2.22: (a) Esquema de partición del árbol octal u octree [Can02]; (b) y (c) ejemplos de representación de sólidos mediante árboles octales.

de la caja envolvente del volumen a codificar [Sam88]. Estas divisiones están hechas de manera que los octantes obtenidos en cada nivel de la descomposición son del mismo tamaño, repitiendo el proceso hasta que las propiedades representadas en todos los nodos del árbol sean homogéneas o bien hasta un nivel de profundidad predefinido (figura 2.22).

Árbol BSP (Binary Space Partitioning).

Se trata de una estructura de datos que representa una división jerárquica y recursiva del espacio tridimensional en otros subespacios. El árbol BSP trata cada polígono 3D como un plano de corte que divide y etiqueta al resto de polígonos como pertenecientes a la parte delantera o trasera de este plano. En otras palabras, cuando se añade una partición al árbol, primero es etiquetada con respecto al nodo principal y a continuación de forma recursiva con respecto a cada una de las ramas del árbol (figura 2.23). Sus aplicaciones más habituales son: eliminación de superficies ocultas en la representación 3D en tiempo real, trazado de rayos (ray tracing), modelado de sólidos, planificación de rutas para robots móviles.

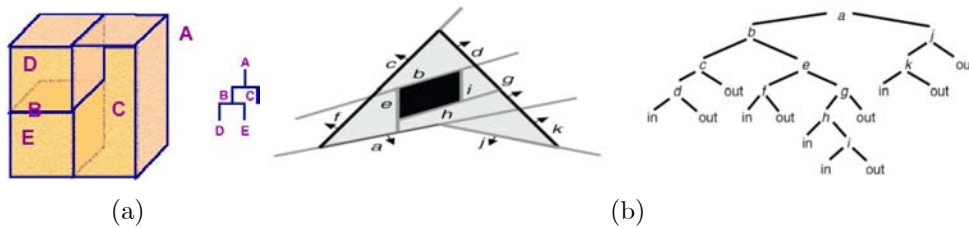


Figura 2.23: (a) Esquema de partición de un árbol BSP en el espacio 3D; (b) visualización de árbol BSP para un polígono en 2D, con su respectiva separación entre lo que queda dentro (*in*) y fuera (*out*) de cada partición.

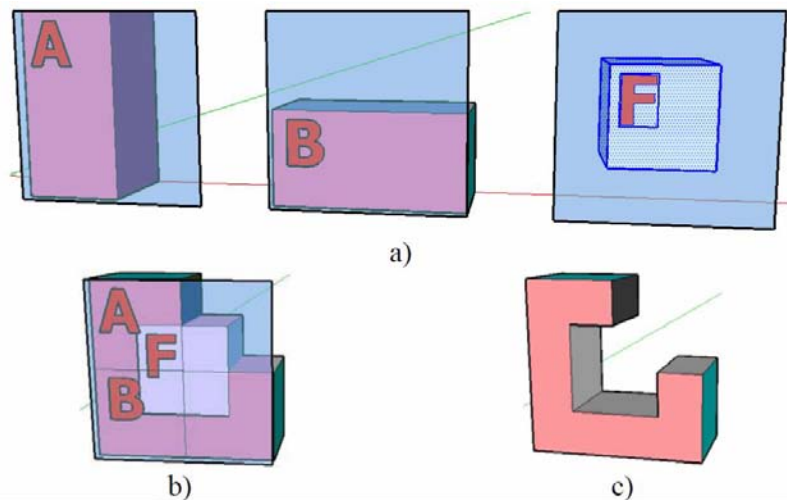


Figura 2.24: Representación CSG: (a) Primitivas con sistemas de referencia; (b) Primitivas en un mismo sistema de referencia; (c) CSG resultante.

Geometría constructiva de sólidos (*Constructive Solid Geometry, CSG*)

La representación mediante este esquema consiste en capturar un proceso de construcción que defina al sólido por medio de operaciones que combinan primitivas [Ros99]. Un objeto es representado generalmente como un árbol binario, ordenado con operadores en los nodos terminales y primitivas en las hojas. Los nodos no terminales pueden ser operaciones booleanas regularizadas (unión, intersección, diferencia) o bien operaciones básicas como traslación, escalado, rotación, etc.

Para generar una representación el esquema hace una lectura ordenada del árbol binario y ejecuta las operaciones entre los objetos para reconstruir un sólido (figura 2.24). Los árboles binarios de este esquema son concisos, no ambiguos, y no son únicos, ya que puede existir otro árbol que genere la misma figura y que por ende pueda existir algún tipo de redundancia.

Sweep

Se define el sólido mediante el movimiento de una forma básica a lo largo del espacio. Estos movimientos pueden incluir una operación de escalado de forma paralela y con una función de crecimiento determinada. En la figura 2.25 pueden verse varios ejemplos de sólidos simples generados a partir de un cuadrado y de un círculo.

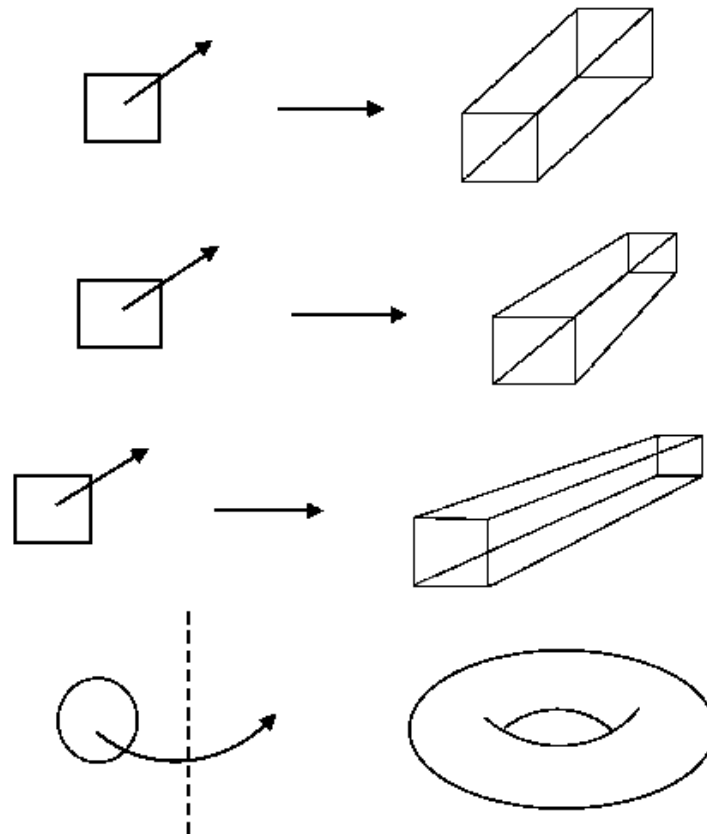


Figura 2.25: Modelado de sólidos mediante la técnica de barrido o *sweep*.

2.3.4. Estructuras de alto nivel

Grafo de la escena

Un grafo de escena es un grafo dirigido acíclico de nodos que contiene los datos que definen un objeto y controlan su proceso de representación. Contiene descripciones de bajo nivel de la geometría y la apariencia visual de los objetos, así como descripciones de alto nivel referentes a la organización espacial de la escena, datos específicos de la aplicación, transformaciones, etc. Los grafos de escena almacenan la información del escenario virtual en diferentes tipos de nodos. Existen nodos que almacenan la información geométrica y actúan como nodos hijos dentro del grafo de escena; el resto de los nodos suelen aplicar algún tipo de modificación sobre el segmento de jerarquía que depende de ellos, bien sea estableciendo agrupaciones, aplicando alguna transformación afín o realizando algún tipo de selección sobre alguna de sus ramas hijas. El proceso de dibujado consiste en realizar un recorrido de dicho grafo, aplicando las operaciones indicadas por cada tipo de nodo. El Grafo de Escena tiene como funciones principales:

- Contribuir a establecer una organización lógica de la escena.
- Establecer dependencias jerárquicas entre distintos sistemas de referencia.
- Posibilitar el proceso de selección entre múltiples niveles de detalle.
- Posibilitar el proceso automático de Culling (eliminación automática de los objetos que se encuentran fuera del campo de visión).

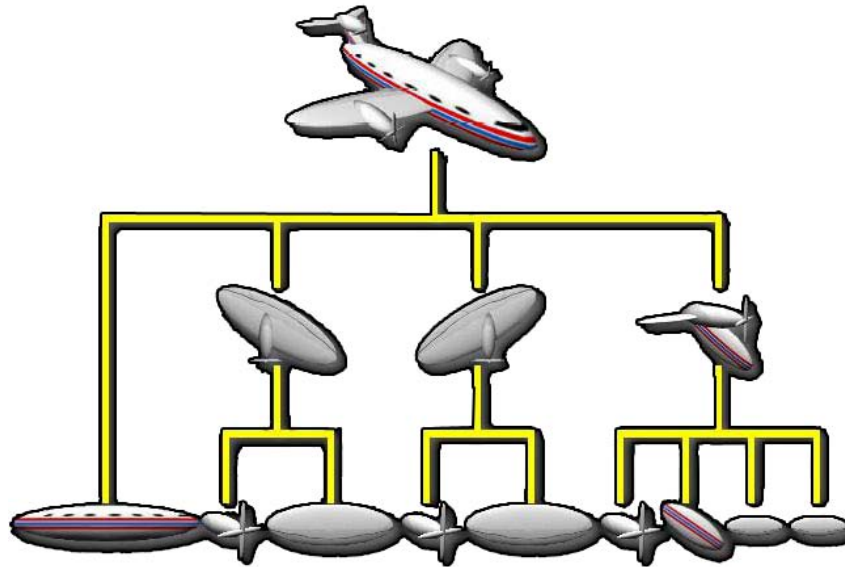


Figura 2.26: Descomposición de un objeto en las diferentes partes que lo componen. El gráfico resultante formaría parte del grafo de la escena en donde se encuentre el objeto.

- Facilitar el control de la escena por parte del usuario.
- Hacer más cómodo el acceso a las librerías gráficas de bajo nivel.

En la figura 2.26 se puede apreciar la descomposición de un objeto en sus diferentes componentes, de manera agrupada, lo cual sería una aproximación al grafo de escena que lo definiría.

Esqueletos (*Skeletons*)

El grafo de esqueleto de un objeto 3D es una representación abstracta geométrica y topológica de su forma tridimensional. Este grafo representa la relación espacial entre las partes geométricas significativas y una estructura de grafo. Cada arco de este grafo representa una parte del objeto con un diámetro aproximadamente constante y aproxima su línea central. Dicho de otra forma, un esqueleto se define como el lugar de los centros de las esferas o discos máximos inscritos en un volumen 3D (figura 2.27). Este grafo es suficientemente conciso y expresivo para utilizarlo como representación abstracta del objeto 3D. La principal dificultad de este tipo de representación es conseguir una interpretación correcta de la forma del objeto.

Sus aplicaciones principales son variadas: catalogación y animación de personajes animados, navegación virtual, planificación de laparoscopias, etc.

2.4. Procesado

Hasta este punto se ha modelado el objeto 3D a partir de la integración de varias vistas parciales del objeto real, obtenidas mediante un escaner 3D. El modelo que se ha obtenido

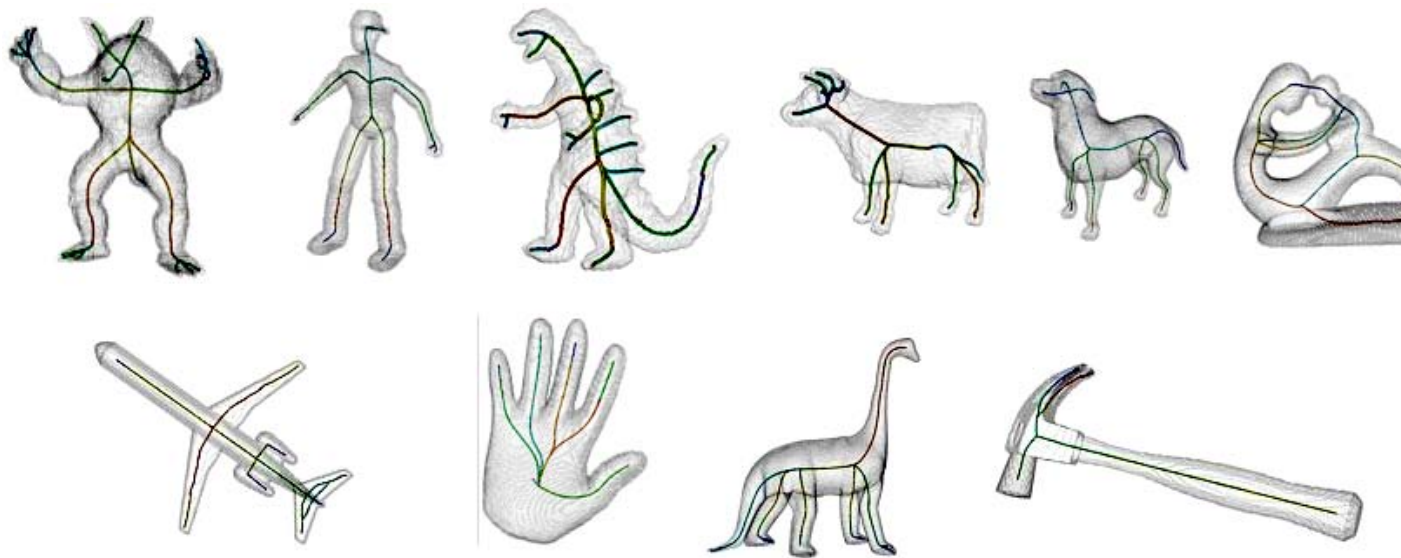


Figura 2.27: Ejemplo de extracción del esqueleto de objetos de formas diversas ([Cor07], [Pan10]).

no es del todo fiel al objeto real debido a que se producen diversos errores en cada una de las etapas del proceso de modelado. Hay que tener en cuenta que para que sea de utilidad en la práctica, un modelo poligonal tiene que satisfacer determinados criterios según la aplicación a que vayan destinados. A pesar de esa dependencia de la aplicación final, se diferencian dos criterios que son comúnmente demandados para el tratamiento de modelos poligonales:

- Criterios de corrección intrínsecos al modelo poligonal: corrección geométrica. Los polígonos han de representar la superficie exterior de un sólido 3D. Esto es, la superficie poligonal ha de ser una variedad bidimensional (sección 2.3.2), es decir, sin puntos singulares donde la superficie se interseque consigo misma o se abra en varias hojas, cerrada y libre de intersecciones.
- Criterios de corrección extrínsecos al modelo poligonal: corrección topológica. Los modelos poligonales han de tener la misma topología que el sólido que representan. En particular, las características topológicas, tales como asas y componentes conectados, han de ser preservados.

La corrección geométrica es particularmente importante en ingeniería y fabricación, donde los objetos sólidos son requeridos para cálculos numéricos, tales como el análisis de los elementos finitos, o para la producción real, tales como prototipado rápido. El requisito de que sea una variedad bidimensional es crítico para el cálculo de medidas diferenciales en las superficies, tales como normales y curvaturas. Por otro lado, la corrección topológica asegura que el modelo poligonal no introduce una complejidad extra con respecto al objeto que representa, tales como asas redundantes y piezas desconectadas, que complicaría de forma innecesaria tareas tales como la simplificación, parametrización y segmentación.

En esta etapa se procesará la malla poligonal bien de forma manual o bien mediante la aplicación de algoritmos automáticos con el objetivo de corregir los diferentes errores que puede contener, para que, de esta manera, satisfaga los dos criterios expuestos. A continuación se enumeran algunos de los errores que se presentan habitualmente:

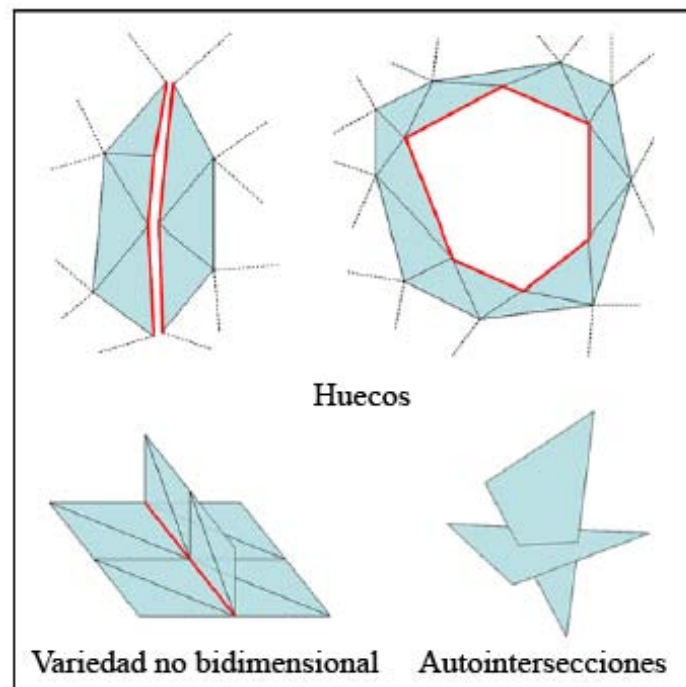


Figura 2.28: Diversos tipos de errores habituales en mallas poligonales. [Ju09]

- *Autointersecciones entre polígonos* que pueden producirse tras el proceso de registro e integración, o bien tras la aplicación de alguna operación de procesado.
- *Normales erróneas*, que se refiere a la existencia de algún cambio en la dirección de las normales, que sea inconsistente con la dirección de las mismas en el resto de la superficie.
- *Elementos que no son variedad bidimensional* (elementos *non-manifolds*): lo que da lugar a que la malla no sea una variedad bidimensional, tales como aristas que son compartidas por más de dos polígonos.
- *Huecos* que producen que la malla poligonal tenga agujeros que no presenta la superficie real del objeto.
- *Picos* (*peaks*, en inglés) o, lo que es lo mismo, caras triangulares que sobresalen de la superficie, con una longitud de arista muy superior a la media, debido principalmente al ruido de los datos de entrada.

Es habitual que las mallas obtenidas en el proceso de reconstrucción no presenten solo un tipo de estos errores sino una combinación de ellos. En la figura 2.28 pueden verse algunos de los errores que se acaban de enumerar. En [Ju09] se presenta una recopilación de técnicas para la corrección de los mismos.

De todos los errores descritos, la aparición de huecos es crítica a la hora de conseguir que una malla sea hermética (*watertight* en inglés) o, lo que es lo mismo, que exista una conexión entre todos los polígonos de forma que esté totalmente cerrada (independientemente de las características topológicas del objeto). Este tipo de mallas son de especial interés para multitud de aplicaciones tales como: sistemas de prototipado rápido, sistemas de metrología,

modelos 3D para aplicaciones de visión por computador o para gráficas por computador, etc. Para conseguir mallas totalmente cerradas es, por tanto, de vital importancia la fase de relleno de huecos.

El tema de la presente tesis se centra en los métodos de relleno de huecos por lo que se procederá a continuación a explicar, con más detalle, este error habitual y los diferentes métodos que pueden encontrarse en la literatura científica para acometer su relleno.

2.5. Métodos de relleno de huecos.

Existen dos hechos que hacen que aparezcan huecos en una malla. El primero es que en el proceso de registro y/o integración se haya producido alguna pérdida o redefinición incorrecta de la malla, por ejemplo, algún triángulo que falte por definir. Este tipo de huecos no suponen un problema y suelen rellenarse de forma directa sin tener en cuenta la información geométrica que rodea al hueco. El segundo hecho es la falta de información geométrica debido a oclusiones o sombras en los objetos. En este caso los huecos suelen ser de mayor tamaño que los anteriores. Además, cuando se trabaja con objetos de forma libre, con cierto nivel de detalle, pueden encontrarse huecos de formas muy complejas, y su relleno se convierte en un problema de gran dificultad. Es por esto que, aunque su solución haya sido ampliamente estudiada, no existe aún un procedimiento general para todas las posibles situaciones que se pueden presentar. Por ello se requiere normalmente de la intervención del usuario para poder evaluar el problema y no existe un algoritmo automático que le dé solución.

El tipo de huecos que se presenta va a depender, por tanto, de las características de la superficie tales como su curvatura, resolución, topología, etc. El grado de dificultad del relleno será mayor cuanto mayor sea la curvatura, la resolución, la profundidad de determinadas cavidades, etc.

El problema del relleno de huecos puede dividirse en dos subproblemas:

1. La identificación de los huecos
2. La creación de geometría que rellene el hueco existente.

En cuanto al primer subproblema, la identificación depende del tipo de representación con que se esté trabajando. En la presente tesis se ha trabajado con mallas poligonales triangulares, mallas a las que se le exige que sean una variedad bidimensional y que estén libres de errores. En este tipo de mallas, la identificación es relativamente sencilla. Dado que la malla está formada por triángulos que comparten cada una de sus aristas sólo con un triángulo vecino, se deduce que los triángulos que estén en el borde disponen de una arista no compartida con ningún triángulo. Partiendo de esta condición, se realizará una búsqueda en la estructura de triángulos de la malla de conjuntos de aristas no compartidas, y se determinarán los caminos o ciclos cerrados que formen estas aristas, y que darán lugar a los bordes de hueco.

En cuanto al segundo subproblema de generación de geometría para rellenar el hueco, puede intuirse que no es un problema trivial dada la complejidad y diversidad de huecos que pueden presentarse, no existiendo ningún método que pueda rellenar automáticamente cualquier hueco que se presente. La mayoría de los métodos existentes tienen dificultades en rellenar huecos de gran complejidad y curvatura elevada. De esta forma, conocido el problema, es posible elucubrar una lista de criterios que debería cumplir un algoritmo de relleno de huecos para que fuese robusto, como la que se presenta en la siguiente sección.

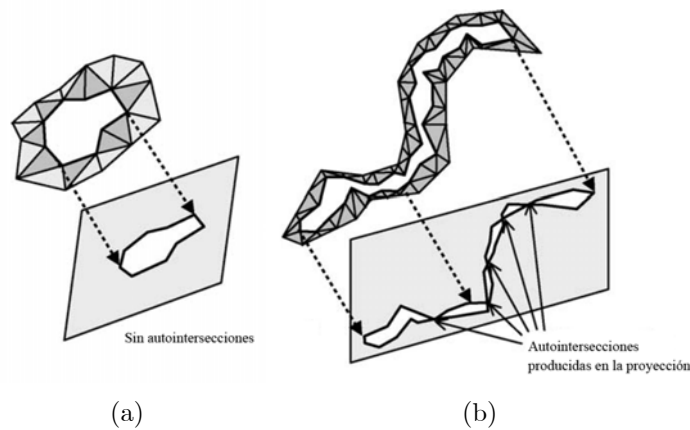


Figura 2.29: Proyección de huecos poligonales en un plano: (a) Hueco simple; (b) Hueco complejo [Jun05].

Pero antes del relleno, puede establecerse una clasificación de los huecos en dos categorías principales, según el tipo de borde de hueco que se presente: huecos simples y huecos complejos [Jun05]. Los primeros son aquellos que pueden ser rellenos a partir de una triangulación plana, dado que todas las aristas pertenecientes al borde de hueco pueden ser proyectados en un plano sin autointersecciones (figura 2.29(a)). La idea principal de la triangulación plana es rellenar huecos con triángulos planos en el plano donde el borde de hueco ha sido proyectado. Con respecto a los segundos, los huecos complejos, debido a la complejidad de su forma no es posible aplicar una triangulación plana y en ellos, los bordes de hueco no pueden ser proyectados en ningún plano sin autointersecciones (figura 2.29(b)).

2.5.1. Criterios de robustez del algoritmo de relleno de huecos

Un buena recopilación de criterios para definir un algoritmo de relleno de huecos robusto es la que propone Podolak en [Pod05]:

1. Debe producir una malla hermética sin autointersecciones, y que sea una variedad bidimensional.
2. Debe tener la capacidad de procesar huecos de forma y tamaño arbitrarios.
3. Debe evitar cambios, aproximaciones o remuestreos de los datos originales. Es esencial que los datos de partida permanezcan invariables a lo largo del proceso de reconstrucción, ya que son los datos reales medidos físicamente mediante sensores.
4. Debe tener la posibilidad de que el usuario incorpore limitaciones para permitir la selección de múltiples soluciones topológicamente diferenciadas. A menos que se desarrollase un algoritmo con elevado nivel de inteligencia, en muchas ocasiones se presentará una solución múltiple para una malla de entrada, por lo que es necesaria la colaboración del usuario.
5. Debe poder procesar mallas escaneadas de gran tamaño y resolución con un tiempo de ejecución que sea proporcional al tamaño de los huecos, en lugar de a la malla de entrada.

6. Debe distinguir a la salida del algoritmo entre las superficies reales observadas y las superficies fabricadas. Este requisito es especialmente importante para determinadas aplicaciones científicas, tales como fabricación, estudio y simulación de piezas de patrimonio histórico, etc.

2.5.2. Clasificación de los métodos de relleno de huecos.

Existen multitud de posibles clasificaciones de los métodos de relleno de huecos, en función de las distintas características diferenciadoras que se tengan en cuenta.

La clasificación en la que se basará la revisión de técnicas de relleno, propuesta en este artículo, será en base a las siguientes dos categorías:

- Métodos en donde el relleno es un proceso implícito en la creación del modelo 3D. Se lleva a cabo de forma implícita durante la fase de integración y reconstrucción de la superficie del objeto.
- Métodos en donde el relleno es un proceso independiente a la creación del modelo 3D.

En los primeros la entrada al algoritmo son las coordenadas 3D de la superficie del objeto sin ninguna información topológica asociada. A partir de estos datos se crea una representación (malla, Bezier, funciones implícitas...) que, por definición, no presenta huecos.

El segundo grupo de métodos de relleno de huecos es más empleado que el primero. La razón es que de esta forma se independiza el proceso de modelado 3D del proceso de rellenado de huecos, permitiendo procedimientos de trabajo más flexibles.

Dentro de este segundo tipo, se puede realizar otra clasificación sería teniendo en cuenta el grado de intervención del usuario. De esta forma podrían diferenciarse entre:

- Métodos automáticos, los cuales realizan el rellenado de forma autónoma sin requerir la intervención del usuario.
- Métodos semiautomáticos, en los que se requiere que el usuario fije determinados parámetros para que el proceso produzca los resultados idóneos.
- Métodos manuales, en los que el usuario que el usuario ha de realizar de forma interactiva el relleno del hueco, especificando las caras que unirán los vértices del borde del hueco.

Casi todos los métodos propuestos intentan que sus algoritmos funcionen de forma automática, aunque suele exigirse que la malla cumpla determinados requisitos impuestos en el proceso de diseño del algoritmo. Como ejemplo de método que requiere una participación directa del usuario, en [He06] se propone un método semiautomático de relleno de huecos en mallas triangulares, mediante el uso de un sensor háptico. Con las técnicas de interacción háptica el usuario puede manipular puntos y triángulos en 3D de forma más intuitiva para modificar y reparar modelos con agujeros complejos.

A continuación van a exponerse diversos métodos de relleno que pueden encontrarse en la literatura científica.

2.5.3. Revisión técnicas de relleno de huecos.

Métodos implícitos.

Del primer grupo de métodos, en donde el relleno se realiza implícitamente con la creación del modelo 3D, pueden encontrarse diversos ejemplos en la bibliografía. Por ejemplo, en [Cur96] convierten cada malla en una función de distancias con signo cuyo nivel de cero es la superficie observada. Seguidamente fusionan estas funciones de distancias y extraen el conjunto de valores nulos, que se corresponde con la superficie final. Para rellenar los huecos, primero marcan como vacía la región del espacio 3D que se encuentra a lo largo de la línea de visión entre el escáner y las mallas. A continuación extraen el borde de esta región como una superficie adicional. El relleno lo hacen creando una superficie que limite la máxima región del espacio que sea consistente con las vistas parciales, por lo que garantiza que se produce una malla hermética.

Otro tipo de métodos son los basados en nubes de puntos, que tratan la unión de todas las vistas parciales como un conjunto de punto 3D no organizados a los cuales hay que ajustar una superficie continua. Al no tener conectividad entre las vistas, los huecos que presentan son conceptualmente equivalentes al espacio entre vistas adyacentes, por lo que estos métodos rellenan los huecos durante la reconstrucción.

Una clase de métodos basados en nubes de puntos interpolan los datos originales utilizando formas alfa (*alpha shapes*) ([Ede94], [Baj95]), cortezas ([Ame98], [Dey01]) o esferas ([Ber99]). Sin embargo, la interpolación puede resultar inadecuada para datos con demasiado ruido. En [Ber99] solucionan este problema aplicando un suavizado previo y consiguen una unión adecuada, sin saltos entre las esferas, aplicando varias veces el algoritmo mientras que se incrementa el radio de las esferas.

Una tercera clase de métodos ajusta un conjunto de funciones de base radial (RBF, *radial basis functions*) a los datos. Se realiza una suma ponderada de esas funciones para dar lugar a una nueva función global, de la que un determinado nivel de valores generará la superficie ([Din01], [Car01]).

Métodos independientes

Respecto al segundo grupo de métodos, en los que el relleno es un proceso independiente a la creación del modelo 3D, es donde se engloban mayor cantidad de algoritmos encontrados en la bibliografía. En todos ellos antes del relleno debe realizarse una búsqueda de los huecos y detección de los bordes de los mismos. Si se trabaja con mallas poligonales, como se ha comentado, la identificación de agujeros se realiza buscando los triángulos que tienen al menos una arista no compartida con ningún otro triángulo. Al unir esas aristas cada ciclo o camino cerrado formado se corresponderá con el borde de un agujero. En [Wu08] proponen un método que realiza una búsqueda más rápida de los bordes de hueco en una malla triangular mediante el uso de *kd-trees*.

Para efectuar el relleno de huecos algunas técnicas mantienen la representación poligonal de los datos, mientras que otros transforman todos o parte de los datos de rango en otras representaciones tales como NURBs, funciones implícitas, proyección en imagen de rango, etc. A continuación se describirán brevemente algunas de esas técnicas.

En [Bra06] primero realizan un análisis de los huecos para determinar cuáles han de ser rellenados y cuáles pertenecen a la topología del objeto. Este análisis consiste en el estudio de la torsión de la curva del contorno de cada hueco. Se basa en la idea de que los huecos que pertenecen a la superficie son suavizados y regulares, mientras que los que son

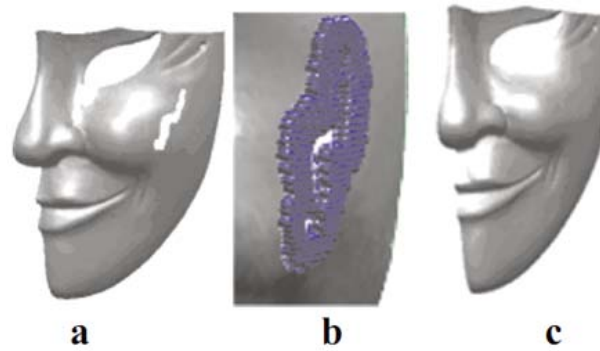


Figura 2.30: Ejemplo de relleno mediante funciones de base radial (RBF). (a) Malla con hueco. (b) Conjunto de centros utilizados para la interpolación. (c) Malla rellenada utilizando interpoladores RBF. [Bra06]

generados por oclusión presentan irregularidades que se traducen en una torsión elevada del contorno. Seguidamente se utilizan los puntos del entorno del borde de hueco para ajustar una superficie mediante interpoladores de funciones de base radial (RBF). Se realizará un proceso iterativo que irá haciendo el hueco más pequeño hasta que se alcance un umbral de ajuste predeterminado.

Una característica importante de este algoritmo es que garantiza que las porciones de malla de relleno se ajustan suavemente en la malla original (figura 2.30). Además, la superficie reconstruida mantiene la resolución de la malla original. Como limitación hay que reseñar que no funciona de manera eficiente con huecos de gran tamaño.

Otro método que utiliza funciones de base radial es el presentado en [Wu08]. Tras la correspondiente detección de huecos, para uno dado, se extraen vértices del borde del mismo y los nodos que tengan un grado de vecindad 2 ó 3 con dichos ellos. Estos vértices se van a utilizar como centros de interpolación para definir una superficie implícita local y que servirá, aplicando las RBF, para interpolar el contenido del hueco. El último paso del algoritmo es la integración de esa porción de malla calculada con el resto de la malla. Dado que el parche de malla sobre el que se han interpolado los puntos de relleno es más grande que el hueco, habrá que distinguir cuáles son los puntos interiores al hueco. Para ello, se realiza una proyección a un plano del parche de malla mediante un análisis de las componentes principales. En el mismo plano se proyectará el polígono correspondiente a los bordes de hueco. En 2D, se determinarán qué puntos quedan dentro del polígono mediante un algoritmo que denominan *algoritmo de comprobación de inclusión de puntos*.

Para terminar se aplica una transformación de 2D a 3D a los puntos de relleno detectados para incluirlos en la malla de partida (figura 2.31).

En [Li10] se presenta un método que emplea técnicas de ajuste polinomial, y que tiene como ventaja la preservación de características de la superficie del entorno del hueco.

En primer lugar, se detectan las características de la superficie en el entorno del hueco (figura 2.32). A continuación se buscan las curvas que se ajusten a esas características superficiales a partir de una minimización dentro de una familia de curvas de ajuste. Estas curvas presentarán discontinuidades en las zonas correspondientes a los huecos. Seguidamente se utiliza una técnica de ajuste polinomial para restaurar las partes incompletas de las curvas superficiales. Para completarlas habrá que asegurar que se satisfaga no solamente la continuidad en la dirección de las tangentes y la curvatura, sino también la continuidad de la torsión.

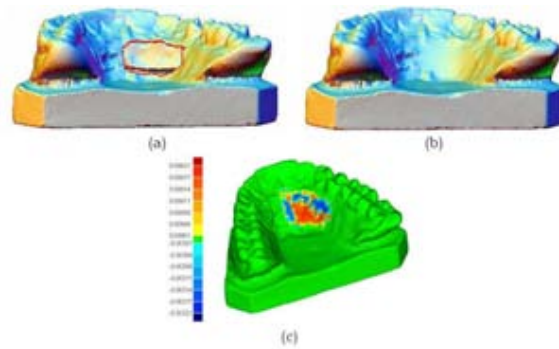


Figura 2.31: Ejemplo de relleno utilizando el método de funciones de base radial. (a) Malla con hueco. (b) Malla reparada. (c) Mapa del error entre la malla reparada y la original.[Wu08]

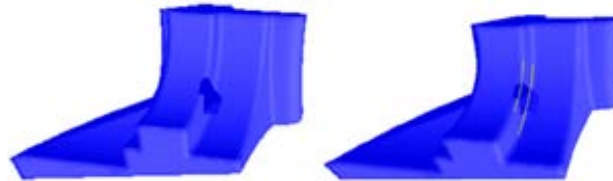


Figura 2.32: Ejemplo de malla con un hueco, y detección de las líneas de las características de la superficie.[Li10]

Las curvas superficiales obtenidas dividen el hueco en otros sub-huecos más simples. A continuación, se determinan todos los puntos de ajuste de cada uno de los sub-huecos y se utilizan parches híbridos de Bézier-Lagrange para rellenarlos. Si el parche de tercer orden de Bézier no se ajusta correctamente o la interpolación de Lagrange no es estable, se requiere de una subdivisión para conseguir el resultado idóneo. Finalmente se lleva a cabo el proceso de triangulación y optimización del parche calculado.

El resultado final es un conjunto de triángulos que se ajusta al hueco y que consigue mantener localmente la forma de la superficie.

El método que se propone en [Bru09] realiza un proceso de despliegue de la malla y, de esta forma, lleva a cabo el relleno en 2D. La aproximación propuesta se lleva a cabo en tres pasos.

En el primer paso, se seleccionan los vértices del borde cerrado del hueco, y se les aplica el proceso de despliegue mediante la minimización de una función de energía. Durante este proceso, se limita el movimiento para que se eviten las autointersecciones. Tras el despliegue, el borde del hueco quedará convertido en un polígono plano simple.

En el segundo paso, se triangula el polígono plano utilizando un algoritmo de triangulación de Delaunay limitado.

En el tercer y último paso, el polígono o parche triangulado es incrustado en el objeto 3D utilizando las posiciones conocidas de los bordes. El parche resultante queda colocado de forma que cierra el hueco. Seguidamente será refinado para que tenga la misma resolución que el entorno de la malla al que pertenece. Los vértices interiores son posicionados de forma que aproximen una superficie de mínima energía (MES, *minimum energy surface*), utilizando dos funciones de energía: una función discreta de energía equitativa [Kob98] y una función de energía laplaciana para obtener una malla de mínimos cuadrados [Sor04]. En la figura 2.33 puede verse un ejemplo de aplicación de este método.

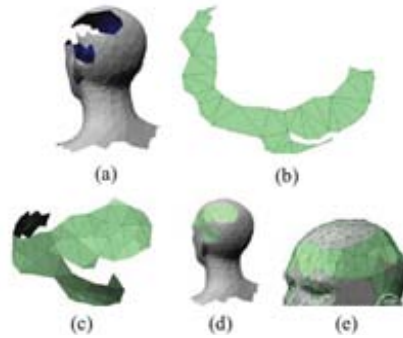


Figura 2.33: Ejemplo de relleno mediante desdoblado de la malla. (a) Hueco en la cabeza del modelo humano. (b) Malla desdoblada. (c)-(e) Malla final obtenida tras la minimización de energía. [Bru09]

Una idea sencilla, pero que ofrece buenos resultados es la propuesta en [Li08]. Aquí se propone un algoritmo que, en primer lugar, divide el hueco en huecos más planos para reducir el número de huecos complejos y para preservar características de detalle de la malla.

Esta división se hace en base a la curvatura asociada a cada vértice del borde de hueco. Partiendo del vértice con menor curvatura, se van añadiendo vértices en un proceso iterativo, y ajustando el conjunto resultante a un polinomio bivariado de grado dos. Cuando la máxima desviación de este ajuste supere un valor determinado, se añade una nueva arista para dividir el hueco, en dos más aplanados.

A continuación, para cada uno de los huecos de la malla, se proyecta el borde de hueco sobre un plano. Si el polígono proyectado tiene autointersecciones, el hueco se definirá como complejo, en caso contrario, será definido como hueco simple.

En el caso de que se presente un hueco simple este será rellenado utilizando una técnica explicada más adelante.

Si el hueco es complejo se hará necesario subdividirlo en otros más simples, utilizando la técnica que ellos denominan “expansión de aristas”. Para ello, se almacenan las aristas que producen la autointersección. En base a una búsqueda de conectividad idónea entre los triángulos cuyas aristas dan lugar a la autointersección, se divide el hueco en otros más simples.

Los nuevos huecos serán sometidos de nuevo al criterio de análisis basado en su proyección sobre un plano. En el caso de que se encuentren de nuevo huecos complejos volverá a repetirse el proceso.

Una vez que todos los huecos complejos han sido divididos en huecos simples, todos los huecos son simples, por lo que es el momento de proceder al relleno de los mismos. Para cada uno, primero se procede a la triangulación del polígono resultante de proyectar el borde de hueco en un plano. Esta triangulación plana se realizará utilizando la triangulación de Delaunay limitada basándose en el Diagrama de Voronoi

A continuación se mapeará esa triangulación plana sobre la malla 3D para así rellenar el hueco (figuras 2.34).

Simultáneamente la resolución de las nuevas triangulaciones puede mejorarse aplicando procesos de subdivisión y suavizado sobre la vecindad de los triángulos ya existentes.

En [Zha07] se aplican dos ideas novedosas para el relleno de huecos en mallas triangulares.

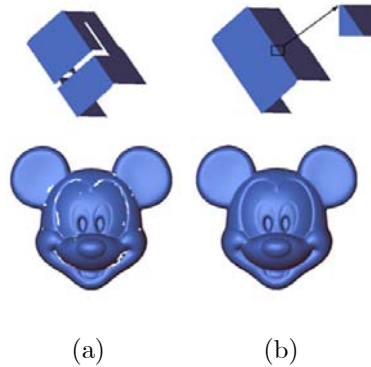


Figura 2.34: Ejemplo de relleno mediante la técnica de subdivisión en huecos más simples [Li08]. En (a) y (b) mallas antes y después del relleno, respectivamente

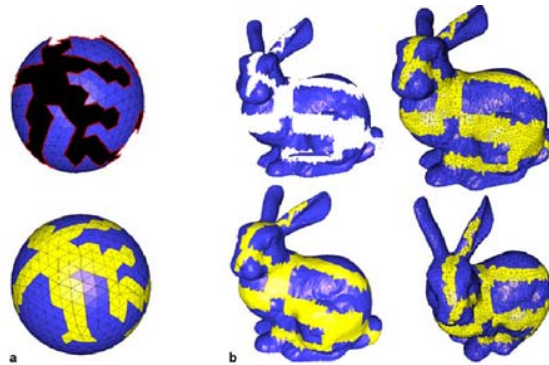


Figura 2.35: Ejemplo de relleno de huecos complejos utilizando la técnica de cálculo de normales.[Zha07]

La primera, la técnica de avance frontal en la malla que es utilizada para generar un cierre inicial del hueco, en lugar de las técnicas tradicionales de triangulación 3D.

La segunda, los triángulos que forman ese parche inicial son modificados mediante la estimación de las normales adecuadas. Finalmente, las tres coordenadas de cada nuevo vértice son recolocadas mediante la resolución de las ecuaciones de Poisson basándose en las normales adecuadas y los bordes del hueco.

Por tanto, en este artículo se presentan dos esquemas de cálculo de normales adecuadas: esquema basado en armónicos y esquema basado en propiedades geodésicas. El primero de ellos se utiliza cuando el entorno del hueco es relativamente plano, en caso contrario se utilizará el segundo esquema.

En la figura 2.35 se puede observar la eficiencia de este algoritmo.

Un ejemplo de aplicación de superficies NURBs es el que puede consultarse en [Kum07]. En este método comienzan cerrando el hueco mediante una malla, basándose en la triangulación de Delaunay. Esta conectividad será la que mantengan finalmente los puntos. A continuación, para cada hueco, se calculan 6 anillos de vértices alrededor del borde de hueco, basados en las relaciones de vecindad. Se calculan las curvas spline que se aproximen a cada uno de esos anillos. Estas curvas se utilizarán para obtener un conjunto de superficie NURBS que cierren el hueco, en concreto 18 superficies suavizadas. Sobre cada una de esas superficies se proyectará la triangulación inicial que se hizo. Las coordenadas finales se calcularán como

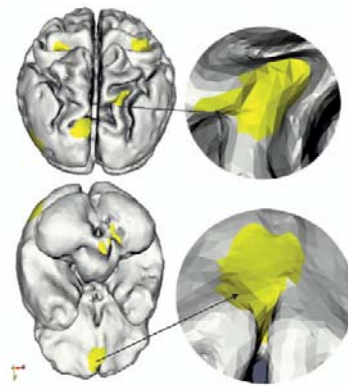


Figura 2.36: Ejemplo de relleno utilizando el método de ajuste de NURBs.[Kum07]

una media entre todas esas proyecciones. Es un algoritmo que solo puede ser utilizado en superficies simples con cierto suavizado (figura 2.36).

Como ejemplo de métodos que utilizan funciones implícitas puede acudir a [Dav01]. Aquí se calcula una función implícita, de distancias, definida en el entorno del hueco a rellenar. Sobre ella se aplica un proceso de difusión que extiende la superficie a lo largo del volumen. Es este un método simple de implementar, que genera superficies que no intersectan entre sí y que es eficiente en grandes mallas o de elevada resolución ya que el algoritmo sólo se aplica en las áreas cercanas a los huecos. En [Guo06] se presenta una variante de este método.

Otro algoritmo muy eficiente en el relleno de huecos y que también utiliza un campo de distancias con signo es el presentado en [Sag08]. En éste se parte de un conjunto de imágenes de rango obtenidas tras la digitalización de un objeto 3D. Para la fusión de esas imágenes de rango y, como representación intermedia, se utiliza un campo de distancias con signo.

El método que se propone genera una superficie para rellenar un hueco mediante la actualización de forma iterativa del campo de distancias, a la vez que se procura que se mantenga la continuidad con las imágenes de rango de partida.

La superficie interpolada se va generando mediante la minimización de la curvatura de la superficie, que es lo mismo que la minimización del área de la superficie interpolada en un dominio local. Dado que la isosuperficie de un campo de distancia puede ser identificada como una superficie interpolada (usando la función distancia), el método propuesto calcula el área de una superficie interpolada, en la vecindad de un voxel, antes y después de invertir el signo de la distancia asociada a dicho voxel. Si el área se ve reducida al invertir el signo, entonces el método propuesto cambia el signo asociado al voxel. Por lo tanto, este método minimiza el área de la superficie interpolada. Este cálculo se va iterando hasta que se llega a la convergencia. Además no se requiere de ninguna condición de borde y el estado inicial del proceso se fija al calcular el campo de distancias con las imágenes de rango de partida. Por otra parte, dado que este método puede ser aplicado a un campo de distancias de resolución adaptativa, se puede decir que funciona eficientemente en agujeros grandes y en entornos de curvatura elevada (figura 2.37).

Otro método que particiona la malla de entrada en volúmenes es el que se presenta en [Pod05]. En este artículo se propone un algoritmo de relleno de huecos en 3D basado en una descomposición del espacio en volúmenes atómicos. El proceso se divide en dos pasos. En el primero, se genera un cubo que contenga a la malla de entrada y se va particionando en volúmenes atómicos. Un volumen es atómico si no intersecta con los polígonos de la malla. A

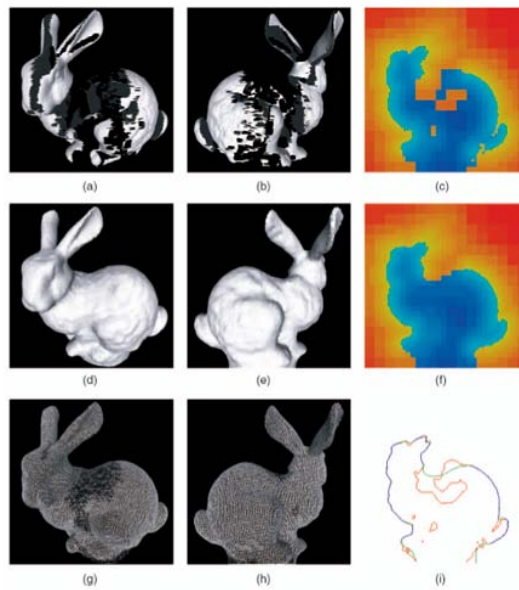


Figura 2.37: Interpolación de huecos grandes en dos imágenes de rango. (a) y (b) son dos imágenes de rango adquiridas de puntos de vista opuestos que ya están alineadas. (c) es una sección 2D del campo de distancias inicial, calculado a partir de las dos imágenes de rango. (d) - (h) resultados tras el relleno de huecos. (i) sección 2D de las mallas generadas.[Sag08]

partir de esta formulación, cada volumen atómico estará complemente o dentro de la malla final de salida (un volumen interior) o fuera de la misma (un volumen exterior). Se define el modelo de salida como la unión de los volúmenes interiores, o lo que es lo mismo, viene implícito al identificar la malla de salida como el borde entre los volúmenes interiores y los exteriores. La definición implícita garantiza que el objeto resultante sea hermético, sin huecos. En el segundo paso, se aplica un proceso de suavizado para hacer menos visible la estructura subyacente de los volúmenes atómicos. En este proceso se fuerza a que la clasificación de los volúmenes no pueda ser modificada mientras que se suaviza, para garantizar que no se produzcan autointersecciones en la malla final.

Nooruddin y Turk en [Noo03] proponen un método que parte de la voxelización de la malla para el relleno de huecos y reparación de zonas mal definidas (dobles paredes, caras

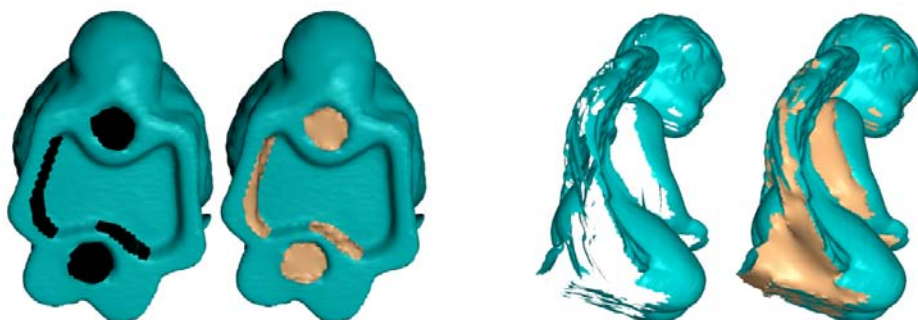


Figura 2.38: Ejemplo de relleno de huecos utilizando la división en volúmenes atómicos propuesta en [Pod05].

con intersección...). Además, basándose en la misma representación, proponen un método que posibilita simplificar la malla. El procedimiento resultante no permite rellenar huecos en zonas de alta curvatura.

Wang y Oliveira en [Wan07] toman 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, se toma un grupo de puntos vecinos a esos bordes y se realiza una interpolación mediante la técnica MLS (Moving Least Squares) [Lan81].

En cuanto a métodos que se basan en técnicas para fotografías 2D, puede destacarse el artículo [Bru07]. En primer lugar, comienza con un rellenado inicial del hueco utilizando una interpolación geométrica. Esta interpolación realiza en primer lugar una triangulación del hueco sin añadir puntos interiores, y evitando que se produzcan auto-intersecciones. Seguidamente esa triangulación es refinada, añadiendo puntos interiores y basándose en Delaunay, para conseguir una resolución similar al resto de la malla.

Seguidamente esta superficie de relleno es deformada utilizando una aproximación de varias vistas estéreo orientadas a la superficie. Es decir se disponen de varias fotografías del objeto tomadas durante el proceso de digitalización para completar la textura del mismo. Tras aplicar un proceso de matching de las fotografías, se utiliza la información de las mismas para formular un problema de minimización de energía basado en una fotoconsistencia y un suavizado Laplaciano. Para resolver este sistema, se parte de los datos conocidos, que son los bordes de hueco.

El método utilizado para la minimización de energía es el esquema Broyden-Fletcher-Goldfarb-Shanno (LSBFGS) de memoria limitada [Noc00].

Otro método que utiliza algoritmos, normalmente aplicados a imágenes 2D es el propuesto en [Bre05]. Aquí se combinan técnicas de ajuste global de la superficie y técnicas de síntesis de textura. La aproximación propuesta tiene, por tanto, dos partes: en la primera se completa la superficie subyacente utilizando técnicas de geometría simple, como las propuestas en [Del02], [Cas02], [Stu01] y, en la segunda parte, se propaga la textura de la superficie 3D desde la porción vista hacia las zonas que se han completado o creado, utilizando una adaptación de las técnicas no paramétricas de síntesis de textura 2D, propuesta en [Efr99]. El objetivo final es completar la superficie con un acabado realista basándose en la propagación del conocimiento que se tiene de las porciones visibles hacia las porciones invisibles. Este proceso se rige por la limitación geométrica impuesta por el relleno realizado en la primera parte de la aproximación. Aunque el relleno que se obtiene no es una reconstrucción precisa del hueco, cuyo contenido es a priori desconocido, es aceptable para hacerse una idea de la forma real del objeto completo (figura 2.39).

Por último, otro tipo de métodos de relleno es el basado en la información del contexto. La idea está inspirada en los métodos de síntesis de textura comentados anteriormente. Un ejemplo representativo de este tipo de métodos es el propuesto en [Sha04]. En lugar de hacer suposiciones a priori de las partes que hay que rellenar, se analizan las superficies que ya existen, y analizando las similitudes en formas que se repiten, junto con la información de la vecindad, los huecos se van rellenando progresivamente (figura 2.40). El algoritmo propuesto parte de una división del volumen en un *octree*, lo que le confiere diversas características a tener en cuenta:

-Multi-escala: La identificación de los huecos, el análisis de la superficie, las medidas de similitud son aplicadas de arriba hacia abajo, o, lo que es lo mismo, de menor a mayor grado

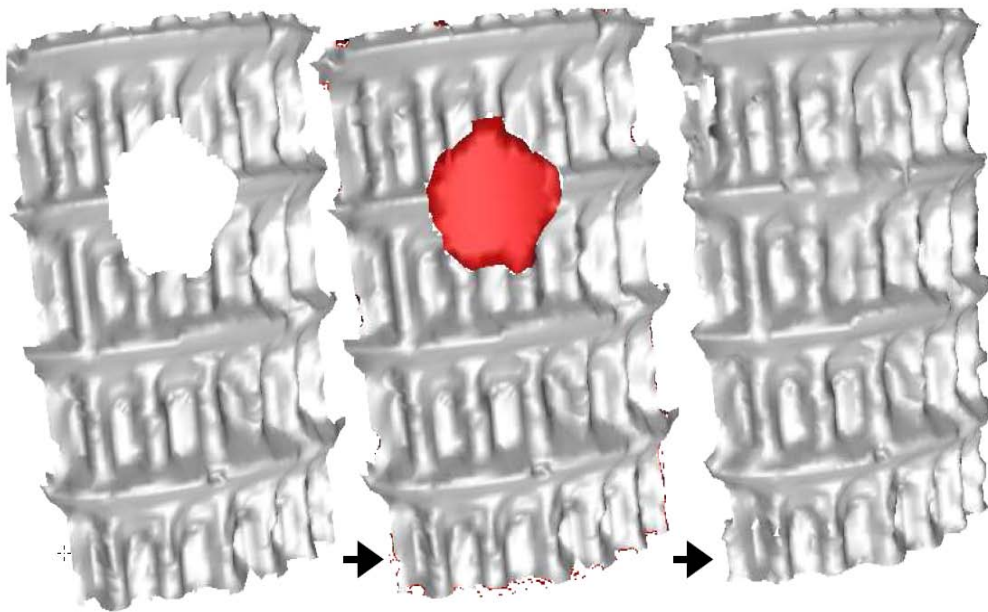


Figura 2.39: Relleno de huecos con síntesis de textura.[Bre05]

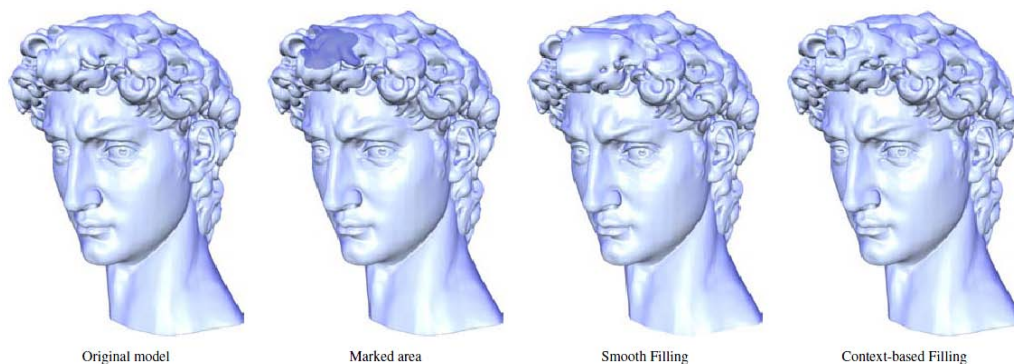


Figura 2.40: Ejemplo de relleno de huecos utilizando un método basado en la información del contexto [Sha04].

de detalle. De forma que los pequeños detalles se infieren a partir de un grado de detalle más bajo.

- Funciones identificadoras o firmas: Se ajustan funciones algebraicas de forma local como una función identificadora de la superficie local que entra dentro de una de las subdivisiones del octree. La firma se utiliza para determinar las similitudes entre conjuntos de puntos.

-Alineación: Para ajustar la porción de malla determinada, con mayor similitud, se aplica una transformación rígida utilizando la técnica ICP junto con pequeñas deformaciones elásticas en funciones asociadas a determinadas subdivisiones del octree.

Capítulo 3

Verificación de los algoritmos de image inpainting en el rellenado de huecos

3.1. Introducción

En el capítulo anterior se explicaban las diferentes etapas que componen el proceso de digitalización de objetos mediante el uso de escáneres 3D.

Como ya se mencionó, el eje principal de esta tesis está fijado sobre la última de las etapas, el procesado y, en concreto, sobre el problema del relleno de huecos

Al final del capítulo II, se ponía de manifiesto que, a pesar de la extensa investigación que existe sobre este problema, aún no existe una solución completa y automática para su resolución.

En esta tesis se plantea la realización de un método que, basándose en algoritmos de restauración de imágenes (image inpainting en inglés), sea capaz de hacer un rellenado de huecos lo más eficiente y robusto posible en mallas 3D. El hecho de emplear estos algoritmos 2D es debido a que suelen generar buenas soluciones en diversas situaciones, en el ámbito de la restauración de fotografías.

Por tanto, en primer lugar, en la sección 3.2, se hará una introducción a las técnicas de image inpainting que pueden encontrarse en la literatura, para seguir a continuación con una explicación más detallada de dos de las técnicas que han sido seleccionadas para la verificación del método propuesto. Seguidamente, en la sección 3.3, se esquematizará el método de relleno que se ha propuesto para continuar, desde la sección 3.4 a la 3.7, con la explicación detallada cada una de las etapas en que se divide dicho método. Para terminar, en la sección 3.8, se planteará la validación del método mediante el análisis de los resultados experimentales obtenidos y en la sección ?? se obtendrán las conclusiones pertinentes.

3.2. Técnicas de restauración de imágenes

Se conoce como *image inpainting* al proceso de restauración de imagen que permite recuperar una parte deteriorada de ésta, o eliminar alguna zona no deseada.

En fotografía digital se trata de traducir a algún algoritmo las técnicas manuales que aplican los restauradores profesionales de cuadros o fotos. Estas técnicas suelen recuperar las regiones dañadas o perdidas a partir de la información que hay disponible en el entorno.



Figura 3.1: (a), Restauración de una imagen antigua deteriorada; (b) Restauración de una imagen estropeada con ruido.

Es por esto que puede ser interpretada como un tipo de interpolación.

Como aplicación típica, en fotografía y cine, de estas técnicas cabe destacar la recuperación de la imagen tras el deterioro del material de soporte. Por ejemplo, la reconstrucción de fotos (como la que se observa en la figura 3.1(a)), eliminación de manchas en la película de cine, etc. Por otra parte, también puede utilizarse para eliminar letras que aparezcan impresas en las fotos (fecha, datos,...), eliminación de ojos rojos, eliminación de objetos no deseados, etc.

Son variados los algoritmos que intentan hacer frente a este problema. En general, muchos de estos algoritmos constan de un paso inicial en el que el usuario ha de seleccionar las partes a eliminar mediante una máscara de selección. Para algunos usos específicos, como la eliminación de ruido, manchas de debidas al polvo, etc,... el algoritmo puede funcionar de forma automática sin la intervención del usuario (véase otro ejemplo en la figura 3.1(b)).

De los diversos algoritmos que pueden encontrarse en la literatura científica, puede proponerse la siguiente clasificación de los métodos propuestos:

1. **Métodos basados en síntesis de textura:** Mediante esta metodología se consiguen rellenar grandes regiones de la imagen con texturas (patrones repetitivos con estocasticidad moderada). En determinados casos se hace un estudio analítico de la textura y en otros se realiza un muestreo de esa textura, copiando y pegando la información en las zonas a rellenar. En el caso de imágenes reales, algunas pueden considerarse formadas por estructuras sin un orden predefinido y otras como una textura con un patrón de repetición. Por tanto, será necesario tener en cuenta esa regularidad o irregularidad en la imagen, lo cual añade complejidad al problema, [?]. Como ejemplos de estos métodos pueden consultarse los trabajos desarrollados en [?],[?], [Efr99], [?] y [?].
2. **Métodos basados en la difusión de información:** Algunos de los algoritmos resuelven el problema específico de eliminación de ruido, polvo, suciedades o texto,..., que no es otra cosa que eliminar y sintetizar pequeñas regiones dentro de la imagen, como los que aparecen en [?], [?],[?], [?], [?]. Estos algoritmos rellenan los huecos mediante la propagación de información de la imagen hacia el interior de la región vía difusión. Estos métodos se basan en las ecuaciones diferenciales que describen el flujo de calor y funcionan con muy buenos resultados, aplicados como algoritmos de restauración. El inconveniente que presentan es que el proceso de difusión introduce cierta borrosidad en las imágenes, que se hace más visible cuanto mayores son las regiones.
3. **Métodos basados en la información estadística:** Otro grupo de algoritmos realizan un estudio estadístico de la imagen a restaurar y/o de otras imágenes similares.

En base a este análisis pueden inferirse los valores de los píxeles que están dentro del hueco que hay que rellenar. En esta dirección, está muy extendido el uso de los Campos de Markov, que consideran la imagen como un campo bidimensional de variables aleatorias.

En el método que se ha desarrollado para esta tesis se utilizaron algoritmos del segundo y tercer grupo para verificar la eficacia del método. Los algoritmos empleados serán explicados más adelante en la sección 3.5. A continuación se iniciará la explicación del método que ha sido desarrollado.

3.3. Método desarrollado. Aplicación de algoritmos de image inpainting al rellenado de huecos en mallas.

Una vez han sido expuestos las diferentes metodologías para el rellenado de huecos en mallas y para la restauración de imágenes, se trata ahora, como se comentó en la introducción, de proponer una idea que combine ambas técnicas: el rellenado de huecos en mallas utilizando los algoritmos de restauración de imágenes. Por tanto, la hipótesis que se plantea, trata de aplicar un método 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 hacer las diferentes transformaciones entre los datos.

El objetivo es emplear un método que se normalmente se utiliza para información bidimensional (imagen) en un modelo 3D que contiene información tridimensional. Para ello podríamos abordar la modificación de los algoritmos de reparación de imágenes para que se pueda aplicar directamente sobre la malla, o, por el contrario, modificar el tipo de representación 3D y adaptarla a 2D, de tal forma que el algoritmo de reparación de imágenes pueda aplicarse sin modificación. En esta tesis se aborda esta segunda forma. Lo que se hará será utilizar una representación bien conocida de representación 3D, que es el de las imágenes de rango, mediante la cual se almacena la información 3D en un formato idéntico a cómo se almacenan las imágenes. Por ello, se procede a convertir el trozo de malla, en donde se ubica el hueco a rellenar, en una imagen de rango, y sobre ella se aplica el algoritmo para rellenar el hueco. Por último, se requiere una conversión de la zona rellena en información 3D, para que el objeto 3D de partida quede rellenado. Este procedimiento que se ha explicado puede ser esquematizado en los siguientes pasos (representados en la figura 3.2):

1. Transformación 3D a 2D para obtener la imagen de rango, I_h , asociada a la malla con huecos M_h .
2. Aplicación del algoritmo de image inpainting. La imagen reconstruida se denotará por I_f .
3. Transformación 2D a 3D y fusión de los datos de relleno con la malla con agujeros. Se transforman los datos de I_f que se corresponden con los huecos rellenos, resultando la malla M_f , que se fusionará con M_h para dar como resultado la malla M' .

Para poder verificar la validez de este método es necesario disponer de un amplio conjunto de resultados experimentales. Para realizar esta experimentación, se ha tomado un conjunto de mallas 3D de objetos que no presentan ningún hueco y a las que se les ha generado huecos artificiales de diversa forma y tamaño (ver Tabla 7.6 en ANEXO XXXXXX). La razón

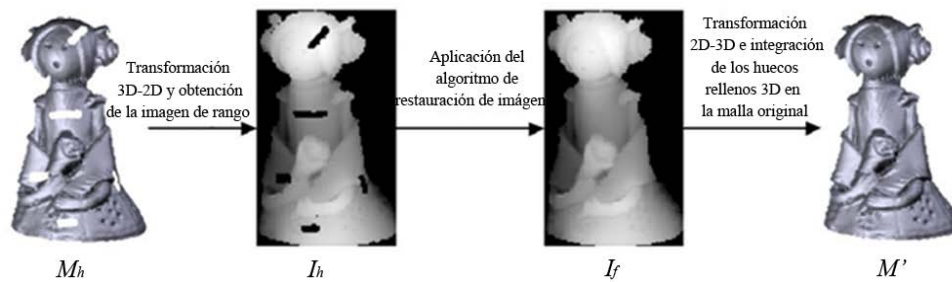


Figura 3.2: Etapas del método de relleno de huecos desarrollado

de proceder de esta manera es para que se disponga de datos que permitan el análisis y verificación de la bondad de los resultados, ya que así puede compararse la malla de partida, sin huecos, con la malla rellena. Por último, en este capítulo se probarán varios métodos de relleno para comparar la viabilidad de cada uno de ellos.

A continuación se procederá a detallar cada uno de los pasos que se han expuesto.

3.4. 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 , de la malla que vaya a ser reparada. Como ya se señaló en la sección ?? del capítulo anterior, la imagen de rango es un método de representación de superficies en el que en una imagen codifica, mediante niveles de grises, la información de altura de los puntos 3D respecto a un plano de referencia o punto de vista. Es evidente que no es válido cualquier plano para obtener una imagen de rango adecuada de la superficie. Su orientación con respecto a la superficie 3D es decisiva para que la proyección resultante sea representativa de la 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 en que se divide el plano de referencia. Esas celdas almacenan las distancias a que se encuentran los puntos de la superficie que les corresponden al proyectar. En un somero análisis del problema, se deduce que si el ancho de esas celdas fuese lo suficientemente grande, varios puntos 3D incidirían sobre la misma celda del plano, dando lugar a una pérdida en la capacidad de representación de la imagen de rango. Por otra parte, si se redujese el ancho de celda demasiado, la nueva situación daría lugar a que cada punto 3D se correspondería con una única celda del plano. Esto evitaría las pérdidas que se daban en el caso anterior pero, sin embargo, daría lugar a otros errores. Ahora se generarían celdas vacías entre las celdas sobre las que inciden puntos consecutivos de la superficie, dada la excesiva resolución de la imagen de rango. Por tanto, la solución ideal sería un punto intermedio entre ambos casos.

Tras lo anterior, se plantea ahora la correspondiente formulación matemática al problema expuesto. De esta forma, el proceso de obtención de la imagen de rango necesita de los dos parámetros fundamentales que se enumeran a continuación:

1. Plano de proyección sobre el que proyectar.
2. Resolución de la imagen de rango.

Para determinar estos dos parámetros se establecen los criterios que se exponen a continuación.

Figura 3.3: Elección del plano de proyección para hacer la transformación 3D a 2D: (a) elección errónea; (b) elección idónea.

Plano de proyección sobre el que proyectar.

En este punto lo que se pretende es elegir qué orientación tendrá que tener el plano de proyección con respecto a la superficie 3D. Es evidente que el plano de proyección que se elija tendrá influencia en la capacidad de representación que tenga la imagen con respecto a los datos 3D. En la figura 3.3 (a) se muestra un ejemplo de una elección errónea del plano de proyección ya que coinciden las proyecciones de dos puntos 3D en un mismo punto de plano, lo que da lugar a una función bidimensional multivaluada. Sin embargo, en la figura 3.3 (b) el plano de proyección elegido no da lugar a situaciones como la anterior.

Como puede fácilmente deducirse, el problema de elección del plano de proyección es diferente si se trabaja con vistas parciales de un objeto 3D que si se trabaja con el objeto 3D completo. En el primer caso, la orientación del plano viene determinada por la posición del escáner con respecto a la superficie o vista parcial. En el segundo caso habrá que encontrar un método de determinación de la orientación idónea para que la imagen de rango sea de utilidad en el método de relleno desarrollado. Este último caso será tratado en el capítulo siguiente.

Inicialmente, para la validación del método de relleno se consideró suficiente el uso de vistas parciales. En el proceso de adquisición el escáner genera las coordenadas 3D de los vértices de la malla con respecto a un sistema de referencia $OXYZ$, centrado en plano imagen y con el eje Z dirigido hacia fuera, es decir, coincidente con el eje de visión del escáner. La elección del plano, en este caso, es inmediata, ya que coincide con el plano XY , que representa el campo de visión del escáner. Esta situación es la que será considerada en lo que sigue de este capítulo.

Resolución de la imagen de rango.

Una vez que se ha determinado la orientación del plano de proyección, el siguiente paso es la definición de la resolución de la imagen I_h en píxeles, para lo cual debe determinarse primero el tamaño de la imagen en milímetros.

Como se dijo anteriormente, dada una vista parcial M_h , si se toman solo las coordenadas X e Y de sus puntos y se descarta la coordenada Z , se obtienen directamente las coordenadas de los puntos proyectados sobre el plano de referencia. De esto se deduce que el tamaño de la imagen es una constante para una malla determinada. Esto es, si la vista parcial que se va a representar, M_h , está formada por C coordenadas, cuyos valores máximos y mínimos en X y en Y son x_{max} , y_{max} y x_{min} , y_{min} , respectivamente, el tamaño que representa la imagen es de $w \times h$, en donde:

$$\begin{aligned} w &= x_{\text{máx}} - x_{\text{mín}} \\ h &= y_{\text{máx}} - y_{\text{mín}} \end{aligned} \tag{3.1}$$

Ésta limita un área rectangular que ha de dividirse en una cuadrícula o rejilla, cuyas celdas se corresponden con los píxeles de la imagen I_h . El número de celdas de la rejilla de proyección determina la resolución y la discretización de la proyección M_h . 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 $(n, m) = (w/q \times h/q)(pixels)$, como puede observarse en la figura 3.4. El píxel (i, j) almacena la información de profundidad, o lo que es lo mismo, la coordenada z_i del nodo $n_i \in M_h$ cuya coordenada (x_i, y_i) verifica

$$\begin{aligned} (i - 1) \cdot q &\leq x_i < i \cdot q \\ (j - 1) \cdot q &\leq y_j < j \cdot q \end{aligned} \tag{3.2}$$

Como se pone de manifiesto, cuanto menor resolución tenga la imagen, mayor será el tamaño de las celdas de la cuadrícula y mayor será el número de nodos M_h que verifican la condición 3.1 o, dicho de otra forma, existirán puntos de la superficie cuya proyección se corresponda con la misma celda de la imagen. En tales casos, se ha optado por almacenar en ese píxel el valor de la menor coordenada z del conjunto de nodos que verifican esa condición. Es evidente que esto conlleva pérdidas de información en la imagen de rango.

En el extremo opuesto, si se elige una resolución de la imagen muy elevada el tamaño del píxel disminuirá, dando lugar a que existan algunas celdas para las que ningún nodo haya verificado la condición 3.1, lo que se traduce en una imagen con zonas vacías. Cuando aparecen estas regiones, se les aplica un algoritmo de rasterización, para que en la imagen de rango exista una continuidad en donde efectivamente es continua la superficie. Este algoritmo, que será explicado en el punto siguiente, conlleva la introducción de datos artificiales, calculados matemáticamente, que no se corresponden con valores reales obtenidos de la superficie. Por tanto, aun disponiendo de esta herramienta, hay que intentar reducir este tipo de regiones vacías en la imagen de rango buscando obtener, a toda costa, una resolución tan óptima como sea posible para evitar la degeneración de la representación 2D de la malla M_h .

Ambas situaciones extremas han de evitarse. Es necesario calcular un valor óptimo de la resolución de la imagen de rango, valor que condicionará la etapa de aplicación del algoritmo de inpainting, o lo que es lo mismo el relleno de huecos que se haga de la malla.

Figura 3.4: Dimensiones de la imagen de rango en función del tamaño del píxel elegido, q .

Figura 3.5: Imágenes de rango en función del ancho de celda elegido

Figura 3.6: Influencia de la uniformidad de la malla en la generación de la imagen de rango

En la figura 3.5 se muestran las proyecciones de una malla M_h , con distintos valores de resolución de la imagen de rango.

En el proceso de generación de la imagen de rango tiene una gran importancia el tipo de distribución de los puntos en la superficie o malla 3D. Así, si la distribución es no uniforme, por más que se pretenda, será imposible encontrar un valor de ancho de píxel que evite las pérdidas mencionadas anteriormente. Con ese tipo de mallas se producirán tanto pérdidas debidas a coincidencias en la proyección de varios puntos sobre la misma celda como pérdidas debido a las regiones de píxeles vacías entre puntos consecutivos de la malla. Dado que la rejilla de proyección es forzosamente de distribución uniforme, interesará que la superficie 3D tenga una distribución lo más uniforme posible (figura 3.6). Esto es inmediato cuando se trabaja con vistas parciales ya que, por el propio funcionamiento del escaner, las vistas son generadas con una distribución uniforme. Sin embargo, en objetos completos la malla final no suele tener una distribución uniforme, ya que son creados mediante vistas parciales, que pueden ser de diferentes resoluciones, y a las que se aplica un proceso de integración y procesado. Es por esto que, al trabajar con objetos completos, se incluirá una etapa al comienzo de remallado de la superficie. Esta etapa será explicada con mayor detalle en el capítulo siguiente.

Por tanto, a partir de lo explicado, queda garantizado que las mallas con las que se trabaja tienen una distribución aproximadamente uniforme. Con este tipo de superficies, el valor adecuado de q estará relacionado con la media de las longitudes de las aristas de la malla 3D. Dado que, de forma general, no se da una uniformidad total y con el objeto de encontrar un valor óptimo de q , se hace necesario buscarlo de forma experimental.

Para determinar la mejor resolución, se ha definido la ocupación óptima por píxel. Así, se dice que un píxel $p_{ij} \in I_h$ está ocupado de forma óptima si existe un y solo un nodo $n_i \in M_h$, que verifique la condición 3.1 para este píxel. A partir de esto, surge la siguiente definición:

Definición 1 *Se define el Ratio de Ocupación Óptima como el ratio entre el número de píxeles ocupados de forma óptima y el número total de píxeles en la imagen. Así, este ratio*

puede expresarse como sigue:

$$O_{I_h} = \frac{\sum_{i=1}^n \sum_{j=1}^n V_{ij}}{n \cdot m} \quad (3.3)$$

Definición 2 donde $n \cdot m$ es el número de píxeles de I_h y V_{ij} es una variable que toma los siguientes valores:

$$V_{ij} = \begin{cases} 1 & \text{si el píxel } p_{ij} \text{ está ocupado óptimamente} \\ 0 & \text{en el resto de los casos} \end{cases} \quad (3.4)$$

De esta manera, O_{I_h} , será pequeño para imágenes de baja resolución. A medida que la resolución se incrementa, el número de píxeles óptimamente ocupados también lo hace, y lo mismo para O_{I_h} hasta el valor en el que cada píxel está ocupado óptimamente. A partir de este punto, O_{I_h} decrece de nuevo, dado que el número de píxeles en la imagen se incrementa mientras el número de píxeles óptimamente ocupados no. El valor de resolución idóneo será aquel que haga máximo a O_{I_h} .

Para fijar el mejor valor del ancho de píxel (que es equivalente al valor de resolución de la imagen), se ha calculado el ratio O_{I_h} para diferentes valores de q_{I_h} para cada malla de la base de datos utilizada. Para que este cálculo sea independiente de las características de la malla, se ha definido q_{I_h} proporcional a la media de las longitudes de las aristas, d_m :

$$q_{I_h} = \alpha \cdot d_m \quad (3.5)$$

donde α es el factor de proporcionalidad y representa la variable relacionada con O_{I_h} .

Por tanto se han ido generando, para cada malla de la base de datos, la variación del ratio de ocupación óptima en función de la variación de α , o lo que es lo mismo de q_{I_h} . Esta relación entre variables puede representarse gráficamente en curvas como las que aparecen en la figura 3.7. En tales curvas se observa que, dado que todas las mallas han sido capturadas con el mismo sensor y, consecuentemente, la distribución de todas es similar, el máximo de todas las curvas se sitúa dentro del mismo intervalo de variación de α .

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. Ésta viene determinada inicialmente por la posición relativa del objeto digitalizado con respecto al escáner, ya que se fijó como plano de proyección el plano XY ($Z = 0$), asociado a la posición del escáner. Por consiguiente, la distancia entre malla y plano de proyección, será, inicialmente, la distancia entre objeto y escáner. Ahora bien, aunque, matemáticamente, la proyección puede llevarse a cabo 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, por tanto, al funcionar los algoritmos de restauración de forma iterativa, se necesitarán más iteraciones para conseguir un valor final adecuado en las zonas de relleno. Por ello, interesará acercar lo máximo posible a la vista parcial hacia el plano de referencia, lo cual se realiza aplicando un factor de escala, f_{esc} , a la imagen de rango como sigue:

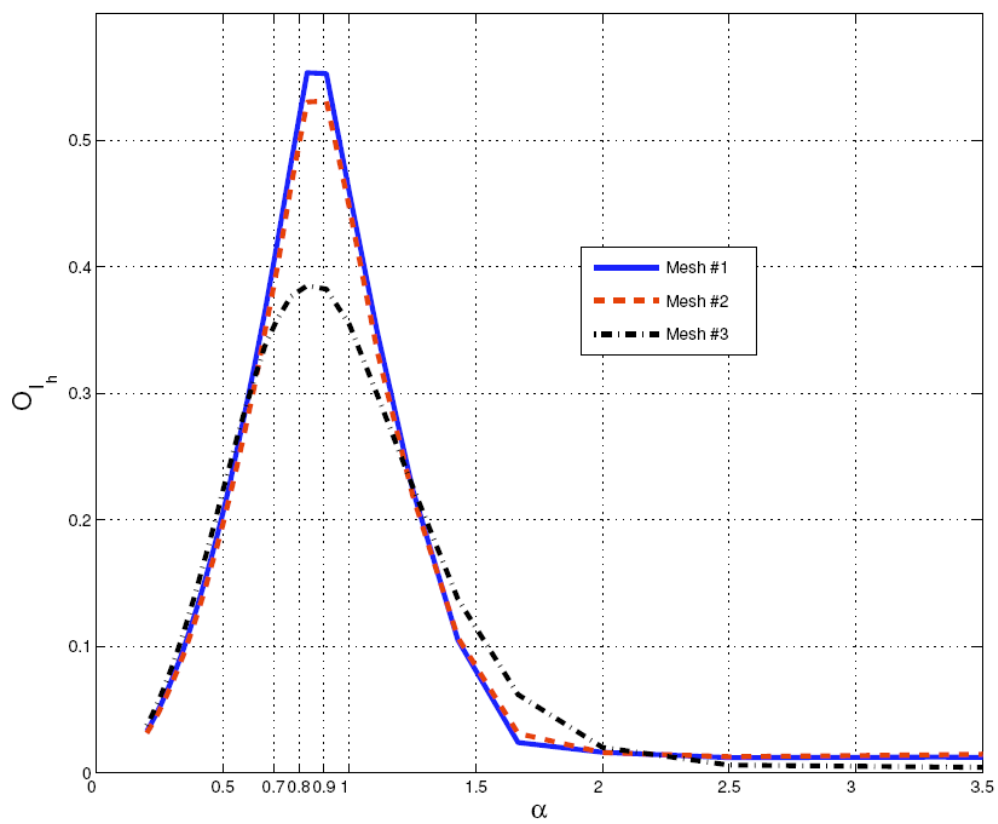


Figura 3.7: Relación entre el ratio de ocupación óptima O_{I_h} y el factor de proporcionalidad α

Figura 3.8: Escalado de los valores, acercamiento en Z al plano Z=0

$$I'_h = I_h \cdot f_{esc} \quad (3.6)$$

En la figura 3.8 aparece representado gráficamente el proceso de escalado. Lógicamente, el factor de escala f_{esc} habrá que tenerlo con los resultados que se obtienen tras el algoritmo de *image inpainting*.

Ahora, una vez obtenida la imagen de rango, I'_h , ya se cuenta con un elemento en formato de imagen sobre el que aplicar el algoritmo de restauración de imagen, y así obtener la imagen restaurada, I_f , que se está buscando. A continuación se van a describir los algoritmos de restauración de imagen que van a ser aplicados.

3.5. Aplicación de los algoritmos de restauración

En esta etapa se va a aplicar el algoritmo de *image inpainting* sobre la imagen I'_h obtenida en la etapa anterior. Como se mencionó anteriormente, una vez revisada la literatura sobre los algoritmos de restauración de imágenes, se decidió comprobar el funcionamiento del método incluyendo dos tipos de algoritmos de restauración, cada uno perteneciente a una de las categorías enumeradas. En concreto, se planteó la aplicación de un método basado en la difusión de información: el desarrollado por Criminisi et al. [?]; y otro método basado en la información estadística: el desarrollado por Roth y Black [?]. En ambos se observó y comprobó la bondad de los resultados que producían al utilizarlos con fotografías, y por eso se decidió aplicarles en el método propuesto.

A continuación se describirá brevemente la base teórica de estos algoritmos para, seguidamente, explicar el modo de integración de esos algoritmos en el método que se pretende verificar en este capítulo.

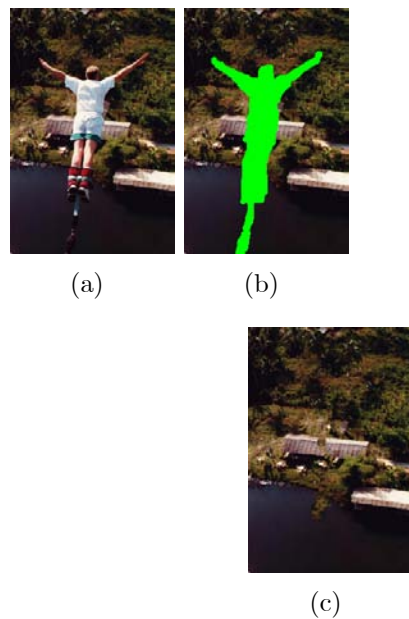


Figura 3.9: (a) Imagen original; (b) Imagen con la máscara de selección aplicada, con la que se señala qué parte se desea eliminar; (c) Imagen resultante tras la aplicación del algoritmo de restauración propuesto por Criminisi et al. en [?].

Algoritmo de restauración de la imagen basado en el muestreo

Este método, descrito en [?], combina las técnicas de síntesis de textura que utilizan muestreo y las técnicas de difusión de imagen para calcular el relleno de huecos en la imagen.

Para este fin, el algoritmo comienza con la introducción por parte del usuario de la máscara de selección. Esta máscara indica qué zonas de la imagen han de ser restauradas. El usuario también tiene que especificar el ancho de la ventana que se utilizará para el muestreo. Aunque en general este parámetro suele fijarse en un valor que funciona bien en la mayoría de los casos.

Durante su ejecución, el algoritmo va rellenando, desde los bordes hacia el interior, los huecos seleccionados. Este relleno lo realiza estudiando la textura y la estructura de la imagen, tomando muestras y propagando la información de textura y estructura hacia el interior de la imagen.

Hay que resaltar que es muy importante el orden en que se realiza el relleno de los píxeles de la imagen. Esto es debido a que se trata de un algoritmo iterativo que va teniendo en cuenta el valor del píxel que se calculó en la iteración anterior. El método que emplean es el llamado *piel de cebolla*, que va rellenando anillos del hueco de forma concéntrica hacia el interior. En la 3.9 puede observarse un ejemplo del funcionamiento de este algoritmo con una imagen en la que se ha querido eliminar uno de los elementos que aparecen.

Algoritmo de restauración basado en el Producto de Expertos

Esta técnica, desarrollada por Roth y Black [?], explota las ideas de la codificación dispersa de imágenes (*sparse image coding*) que consiguen representar parches de la imagen como la

combinación lineal de filtros aprendidos para el aprendizaje del comportamiento estadístico de las imágenes, utilizando los Campos de Markov (MRF- Markov Random Fields).

El método empleado se denomina Campo de Expertos y modela la probabilidad de una imagen en términos de un campo aleatorio con cliques (parches con determinadas relaciones de vecindad) solapados, cuyos potenciales son representados como un Producto de Expertos [?].

La función del Producto de Expertos es modelar distribuciones de probabilidad de elevadas dimensiones (parches de la imagen) a partir del producto de varias distribuciones de expertos, donde cada experto trabaja en un subespacio de baja dimensión, sencillo de modelar. En concreto, al aplicar un filtro lineal a un parche de imagen se produce el subespacio de una dimensión que se modelará. Por tanto, la densidad de probabilidad del parche de imagen puede ser definido como el producto de las distribuciones que describen los filtros lineales.

Para modelar la probabilidad de la imagen completa, se emplea el Campo de Markov que representa la probabilidad de la imagen como una distribución de Gibbs $p = \frac{1}{z} e^{-\sum V_k(x_k)}$, donde x es una imagen y $V_k(x_k)$ es la función potencial para el clique x_k . Considerando un Campo de Markov homogéneo, todos los potenciales V_k son los mismos para todos los cliques, $V_k(x_k) = V(x_k)$. Para definir la función V , los potenciales del Campo de Markov son representados como el Producto de Expertos para todos los cliques. Consecuentemente, la probabilidad de la imagen puede ser modelada en términos del producto de las distribuciones de filtros lineales.

Todos los parámetros en este modelo son aprendidos a partir de un conjunto de imágenes, maximizando su probabilidad, es decir, maximizando la probabilidad para los modelos del Producto de Experto y del Campo de Expertos.

El algoritmo propaga la información utilizando el Campo de Expertos dentro del conjunto de puntos pertenecientes a la vecindad de los píxeles que se desean rellenar. La propagación se realiza de forma iterativa, teniendo en cuenta el modelo de probabilidad de la imagen (modelo de Gibbs). Como puede observarse en la 3.10 llama la atención la eficacia del algoritmo en la restauración de las imágenes.

Este modelo ha sido entrenado utilizando 2000 regiones de imágenes que se han extraído de la base de datos para segmentación de Berkeley Segmentación (escenas naturales, personas, edificios, etc) [?]. En la Tabla 3.1 se muestra un grupo de imágenes pertenecientes a esa base de datos.

Aplicación del algoritmo de restauración a la imagen de rango

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, para la aplicación de estos algoritmos habrá que introducir tanto la imagen de rango, I'_h , como una máscara de selección de los píxeles a rellenar, Q_h . Esta máscara es una imagen binaria, del mismo tamaño que la imagen de rango, $(n, m) = (w/q \times h/q)(pixels)$, y en las que aparecen activos los píxeles sobre los que se pretenda realizar el relleno, como se muestra en la figura 3.11.

CAPÍTULO 3. VERIFICACIÓN DE LOS ALGORITMOS DE IMAGE INPAINTING EN EL RELLENADO DE BUECOS
VERIFICACIÓN DE LOS ALGORITMOS DE RESTAURACIÓN

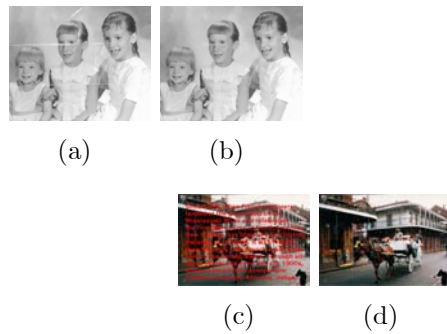


Figura 3.10: (a) Fotografía antigua deteriorada; (b) Restauración de la fotografía antigua mediante la aplicación del algoritmo propuesto en [?]; (c) Imagen original con un texto sobrepuesto que desea eliminarse; (d) Imagen con el texto eliminado tras la aplicación del algoritmo de [?].

Tabla 3.1: Selección de un subconjunto de imágenes del banco de imágenes de Berckley utilizado para el entrenamiento del algoritmo propuesto por Roth et al [?]



Figura 3.11: Ejemplo de Máscara de selección

3.6. Transformación de 2D a 3D. Triangulación de huecos

Para completar la reparación de la malla, es necesario transformar los datos que se han obtenido en la etapa anterior a 3D. Solamente será necesario aplicar esa transformación a los datos nuevos que se han generado tras la aplicación 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, tal y como se muestra en la figura 3.12.

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 cada píxel almacenaba la distancia a la que se encontraba el punto 3D de la malla cuya proyección ortogonal sobre el plano de proyección, incidía sobre la celda correspondiente a dicho píxel. Por tanto, el valor que almacene cada píxel rellenado, indica la distancia a la que se encuentra el punto 3D con respecto al plano imagen, y, en el caso de vistas parciales, se correspondía con la coordenada z . Por lo tanto, una vez aplicado el relleno, se obtienen directamente las coordenadas z de los puntos que formarán el relleno 3D. 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 (r, s) que almacena el valor r , la correspondiente coordenada 3D para ese píxel será:

$$(x, y, z) = \left(x_{\min} + s \cdot q, y_{\max} - r \cdot q, \frac{r}{f_{esc}} \right); \quad r = \{1, 2, \dots, n\}; \quad s = \{1, 2, \dots, m\} \quad (3.7)$$

donde f_{esc} es el factor de escala que se aplicó en la Ecuación 3.6 para acercar los valores de la imagen de rango al nivel de cero en Z (al plano XY). Hay que señalar que en la ecuación

Figura 3.12: Obtención de los píxeles rellenos de la imagen restaurada mediante la aplicación de la máscara.

anterior se ha considerado que las coordenadas de los píxeles vienen definidas por la fila r y columna s que ocupan en la matriz imagen. Es por ello que la coordenada x depende de la segunda coordenada del plano imagen, s , y la coordenada y de la primera, r . En la figura 3.13 puede verse la relación entre las coordenadas del plano imagen y las coordenadas 3D.

Una vez calculadas todas las coordenadas se dispone de una nube de puntos 3D, R_i , correspondientes a los i huecos de la malla que han sido rellenos. Ahora es el momento de definir las superficies rellanadas 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_i} . Aunque estos puntos pertenecen a la malla inicial, M_h , sirven como delimitadores de cada una de las zonas rellenas. A su vez, tendrán la función de servir como eje de unión o cosido entre la malla inicial con huecos, M_h , y las mallas de relleno, H_i . En la figura 3.14(b) se puede apreciar esta diferenciación: en rojo se representan los puntos de los huecos (los devueltos por los algoritmos de reparación) y en azul, los puntos del borde de los huecos (figura 3.14(a)).

En la sección ?? del capítulo anterior se veía la diferencia entre una nube de puntos estructurada y una no estructurada. Pues bien, en el problema que nos ocupa se observa claramente que los huecos rellenos en la imagen se traducen, al aplicar la ecuación 3.7 en una nube de puntos estructurada, cuyos puntos mantienen la relación de vecindad que les venía definida según la posición relativa que ocupaban sus correspondientes píxeles en la imagen. Por tanto, la triangulación de los puntos generados en el relleno, R_i , se obtiene de forma inmediata a partir de las relaciones de vecindad que se extraen de la imagen de rango restaurada (figura 3.15). Esta triangulación, que se denominará T_{1_i} , da lugar a i porciones de malla H_{1_i} , correspondientes a los i huecos que se han relleno, y que no están conectadas con la malla inicial M_h . Por ello, buscando esa fusión entre ambas mallas, se hace necesario

Figura 3.13: Correspondencias entre las coordenadas 3D y las coordenadas del plano imagen.

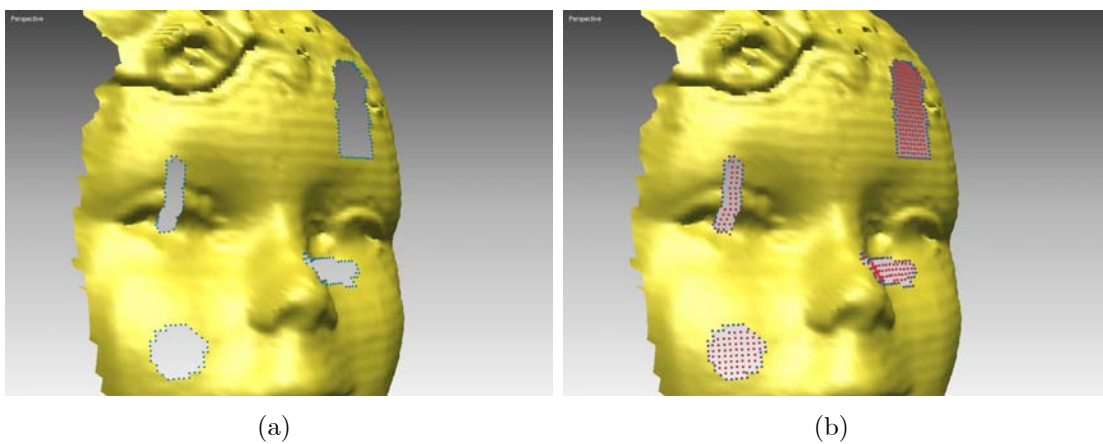


Figura 3.14: Puntos que se van a triangular (los de los huecos –en rojo- junto con los del borde de los mismo –en azul-): (a) Puntos del borde de los huecos; (b) Puntos que se van a triangular.

Figura 3.15: Relación de vecindad entre puntos / relación de vecindad entre píxeles. Nube de puntos estructurada

calcular otra triangulación T_{2_i} entre los bordes de esas porciones de malla, denominados B_{1_i} , y los bordes B_{H_i} de los huecos de M_h .

Tal y como fue definido el proceso de generación de la imagen de rango en la sección 3.4, se fijaba un ancho de píxel óptimo para evitar una degeneración en la representación de la superficie 3D. También se decía que las pérdidas en el proceso tendían a ser nulas cuanto más uniformidad presentase la malla 3D. Dado que no puede asegurarse una uniformidad total, es evidente que siempre se producirá una pérdida de datos en la creación de la imagen de rango. Es decir, no todos los puntos 3D de la malla vendrán representados unívocamente en la imagen de rango. Por tanto, esta afirmación también será aplicable a los puntos que forman los bordes de los huecos B_{H_i} . Es por eso que la triangulación que una los bordes de las porciones rellenas con los bordes de hueco, no puede obtenerse, de forma inmediata, a partir de la imagen de rango, como se explicaba en el párrafo anterior. El cálculo de esta triangulación, T_{2_i} , entre los bordes de las mallas, B_{1_i} , y los bordes de huecos, B_{H_i} , se llevará a cabo en el espacio 3D, con la nube de puntos $B_{1_i} \cup B_{H_i}$, como puede observarse en la figura 3.16.

El método de triangulación elegido ha sido el que se propone en [?]. Este método, que está basado en la triangulación de Delaunay ([?]) extendida a 2.5 D, utiliza un triángulo inicial como semilla del proceso y a partir de él se va extendiendo la triangulación a toda la nube de puntos. La principal ventaja de este método es la rapidez de cálculo en grandes nubes de puntos ya que aplican las ideas de [?] de cálculo adaptativo de punto flotante.

La triangulación final de la nube de puntos $R_i \cup B_{H_i}$ se obtiene como la unión $T_{1_i} \cup T_{2_i}$ y da lugar las porciones de malla H_i . Debido al método de triangulación empleado y ante la diversa complejidad que pueden presentar los bordes de huecos, han de procesarse las mallas H_i resultantes para detectar y corregir los posibles errores. Concretamente, en [?] se menciona que pueden generarse los siguientes tipos de errores:

- • Vértices que no constituyen una variedad bidimensional (vértices *non-manifold*).

Figura 3.16: Esquema de las triangulaciones boundary 1 U boundary 2

- Huecos en la triangulación.
- Aristas que no constituyen una variedad bidimensional (aristas *non-manifold*).
- Puntos, aristas y triángulos exteriores al contorno de limitado por B_{H_i} .
- Triángulos duplicados.
- Triángulos con orientación diferente.

Una vez concluída la triangulación y aplicado el proceso de corrección de errores se tiene un conjunto de i mallas de relleno H_i . En la figura 3.17 aparecen las mallas de relleno correspondientes a la aplicación del método sobre la malla que aparece en la figura 3.14. Es el momento ahora de continuar con la siguiente etapa del proceso en la que se integrarán estas mallas H_i con la malla inicial M_h .

3.7. Integración de los huecos triangulados con la malla inicial

La última etapa del método de relleno desarrollado es la obtención de una malla M' ya restaurada, con los i huecos rellenos. Como ya se mencionó en la etapa anterior, se realizó una triangulación de la nube de puntos formada por la unión entre los puntos generados en el relleno R_i y los bordes de los huecos B_{H_i} , para así conseguir una continuidad tras la unión de las mallas M_h y H_i . Como producto resultante, tras la aplicación del método de relleno, se van a generar dos tipos de mallas:

1. Las mallas H_i que ya fueron generadas en la etapa anterior y que representan las zonas que se han generado mediante el algoritmo de relleno.

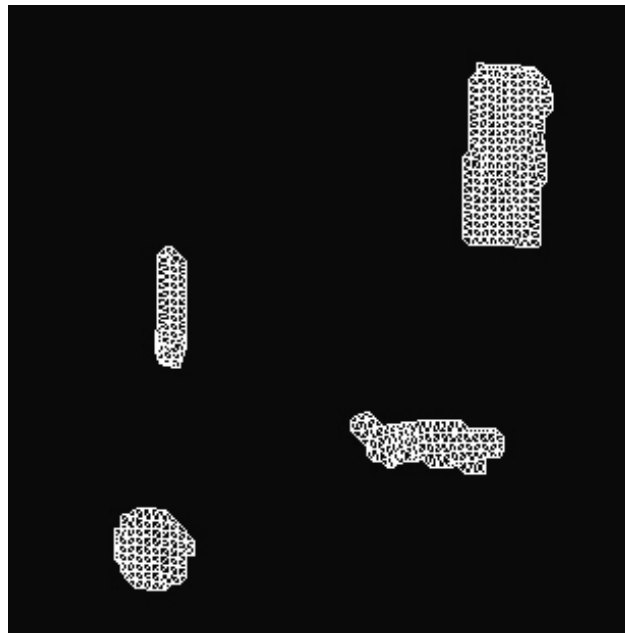


Figura 3.17: Huecos una vez realizada la fase de triangulación.

2. La malla M' , resultado de la unión $M_h \cup H_i$, que constituye la malla restaurada, con los huecos rellenos.

El primer tipo de mallas es interesante para tener la posibilidad de discernir, en todo momento, entre las partes de los resultados obtenidos mediante digitalización y las que han sido generadas artificialmente mediante el algoritmo. El segundo tipo de mallas constituye el producto final del algoritmo, una malla restaurada sin huecos que era lo que se perseguía con la aplicación de este método. Además esta malla será de utilidad para la validación del método desarrollado, ya que podrá compararse con la malla digitalizada inicial, sin huecos generados. Sin embargo, como inconveniente, esta malla no permite diferenciar entre la malla con huecos y las zonas regeneradas artificialmente.

En la figura 3.18 se representan de dos formas los resultados obtenidos tras realizar la fusión.

3.8. Validación del método de relleno propuesto

Como ya se adelantó al inicio de este capítulo para demostrar la validez del método se planteó la utilización de vistas parciales, por la facilidad que ofrecen para la transformación y adaptación de los datos 3D al algoritmo de restauración de imágenes. Además, se estableció que para la experimentación se crearía una base de datos de vistas parciales sin huecos, a las que se les generarían huecos artificialmente de tamaño y forma diversos. El hecho de proceder así, en lugar de utilizar vistas que, tras el proceso de digitalización, presentasen huecos reales era por la posibilidad de contar con una base de comparación a la hora de comprobar la validez del método. Por consiguiente, el disponer de una vista completa, sin huecos, M_I , permite contar con un elemento para la comparación con el resultado M' y para la decisión sobre la validez y robustez del método de relleno que se ha aplicado. Es decir, podría decirse que M_I sería la solución ideal (*ground truth*) que todo algoritmo de

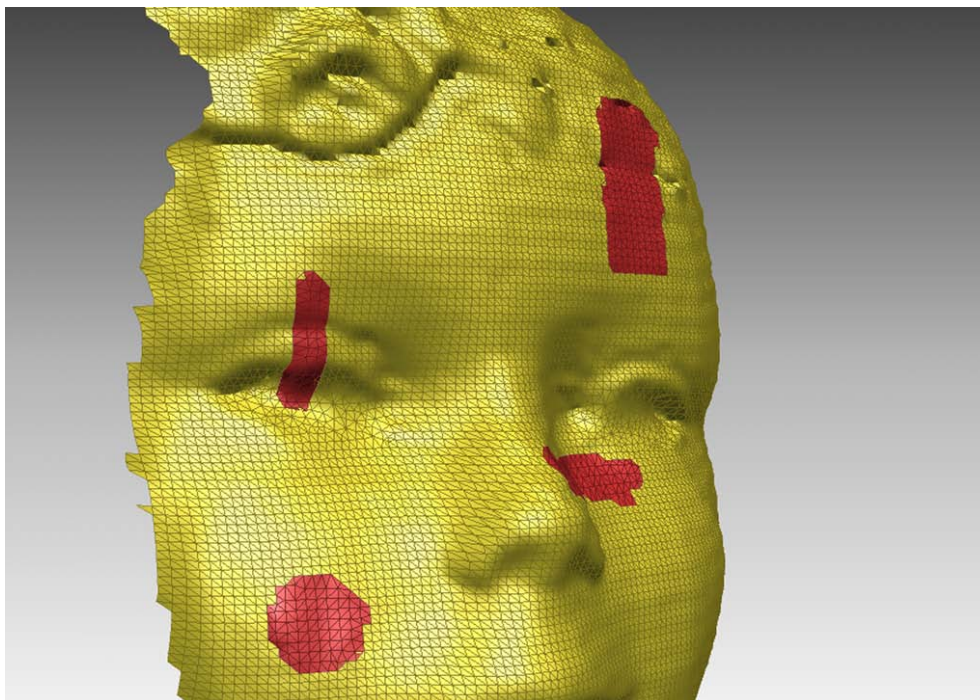


Figura 3.18: Malla final reconstruida, con los huecos coloreados en rojos para observar la reconstrucción

relleno de huecos debe perseguir y , por tanto, se trata de evaluar cuánto se han alejado cada uno de los resultados de la solución ideal.

Por otra parte, se planteó el estudio de las diferencias en los resultados que el método ofrecía utilizando diferentes algoritmos de restauración de imágenes, mencionados anteriormente en la sección 3.5. De esta forma, se aplicó el método de relleno sobre la base de datos de j vistas parciales, usando los algoritmos de [?] y [?], con la consiguiente obtención de $2j$ mallas restauradas. En lo que sigue, se denominará *Algoritmo 1* al propuesto en [?] y *Algoritmo 2* al de [?] y, por lo tanto, las mallas resultantes se denotarán por M'_1 y M'_2 .

Además, para someter a un juicio más riguroso a ese método, también se aplicó a las j mallas M_h un proceso de relleno mediante el software comercial *Rapidform*, que se denominará *Algoritmo 3*, obteniendo así otra malla restaurada M'_3 . De esta manera se establecieron las comparaciones entre la malla original sin huecos M_I y las $3j$ mallas rellenadas M'_1 , M'_2 y M'_3 .

Para poder llevar a cabo la comparación han de definirse qué parámetros son de utilidad para computar las diferencias entre los métodos de relleno aplicados. Un parámetro que surge intuitivamente al intentar evaluar esas diferencias es la distancia entre la solución ideal M_I y las soluciones ofrecidas por cada uno de los métodos. Así pues el problema se reduce al cálculo de la distancia, $d(a, b)$ entre dos mallas triangulares a y b , siendo a la malla inicial M_I y b las diferentes soluciones obtenidas M'_1 , M'_2 y M'_3 . Para calcular esta distancia puede eliminarse, para simplificar, la triangulación de la malla, y puede considerarse, como una buena aproximación, el cálculo de las distancias entre las nubes de puntos que conforman las distintas mallas. Así pues, esto puede expresarse como:

$$d(M_I, M'_X) \approx d(f_m, g_n) = \min d(f_k, g_n) \quad \forall k = 1, 2, \dots, m \quad (3.8)$$

siendo X cualquiera de los métodos de relleno: 1, 2 y S ; f_m los m vértices de la nube de puntos de la malla M_I ; g_n los n vértices de la nube de puntos de la malla M'_X . Lo que expresa la ecuación 3.8 es que para obtener el vector de distancias $d(M_I, M'_X)$ se toma un punto de f_k de f_m , se calculan las distancias de f_k a todos los puntos de g_n y el mínimo de esas distancias es el que se almacena en la primera posición k del vector $d(M_I, M'_X)$. De esta forma, se completa el cálculo de dicho vector procediendo así para todos los puntos de f_m .

Es evidente que solo existirán diferencias entre esas mallas en las zonas correspondientes a los huecos, ya que, como se explicó en el punto anterior, el resultado final del método se obtenía mediante la integración de la malla M_h y las mallas relleno H_i . Es por esto que, en las zonas de la malla que no existen huecos, son totalmente coincidentes con la malla inicial M_I . Esto es también aplicable a las soluciones obtenidas mediante el software comercial, ya que, si bien no es conocido el algoritmo que se emplea, se considera que no son modificadas las zonas sin huecos. Teniendo en cuenta cómo se iban almacenando las mallas rellenas, en las que a continuación de los datos ya existentes se añadían las zonas rellenas, todo esto puede formularse como sigue:

$$\exists k_1 \in \{1, 2, \dots, n\} / d(M_I, M'_X) = d(f_m, g_{n'}) \quad \forall n' > k_1 \quad (3.9)$$

en donde con k_1 se denota el índice de la nubes de puntos g_n a partir del que se almacenan los puntos que sólo se corresponden con los huecos. De aquí surge otra expresión evidente, que redundante en la misma idea:

$$d(M_I, M'_X) = d(f_m, g_{n''}) = 0 \quad \forall n'' \leq k_1 \quad (3.10)$$

De esta forma, el problema real consiste en al cálculo de la distancia $d(M_I, M'_{hX})$ entre la solución ideal M_I y las mallas correspondientes a los huecos rellenos M'_{h1} , M'_{h2} y M'_{h3} .

Para obtener el valor de esta distancias, y, con el fin de evitar un cálculo en bruto de las mismas, computacionalmente muy costoso, se utilizó la representación *Kd-tree* que supone una aceleración considerable de los cálculos. El *Kd-tree* es una estructura de datos para la partición del espacio que organiza los puntos en un espacio euclídeo de k dimensiones. Los *Kd-trees* son un caso especial de los árboles BSP que fueron explicados en la sección 2.3 del capítulo 2. Un *Kd-tree* emplea sólo planos perpendiculares a uno de los ejes del sistema de coordenadas. Además, todos los nodos contenidos en él, desde el nodo raíz hasta los nodos hoja o nodos terminales, almacenan un punto, con lo que cada plano contiene a uno de los puntos del *Kd-tree*. El uso de este tipo de estructuras pueden acelerar enormemente el trabajo con nubes de puntos, de tal forma que, permiten la búsqueda de todos los puntos contenidos en un rango de coordenadas en un tiempo del orden $O(\sqrt[p]{p} + k)$, donde p es el número de puntos del árbol y k el número de puntos que cumplen la condición de búsqueda. Así mismo, la búsqueda del vecino más próximo puede ser llevada a cabo en un tiempo del orden $O(\log(p))$. En la figura 3.19 se muestra gráficamente la representación mediante *Kd-tree*.

Así, si se representa la malla inicial M_I mediante esta representación, se obtiene la estructura K_I . A partir de ahí, para determinar las distancias entre las mallas habría que calcular las distancias siguientes:

$$d(M_I, M'_X) = d(K_I, M'_{hX}) = d(K_I, g_{n'}) \quad \forall n' > k_1 \quad (3.11)$$

siendo X uno de los diferentes métodos de relleno 1, 2 o 3.

Figura 3.19: Kd-tree, distancias entre dos mallas triangulares

Una vez fijada la herramienta de cálculo, hay que determinar qué información puede servir para evaluar los resultados. En primer lugar, se obtuvo para cada una de las $3j$ vistas parciales un mapa de distancias que constituye un método para evaluar visualmente los resultados. El mapa de distancias consiste en una representación visual en 3D de las distancias que han sido calculadas. En concreto, se representan las vistas parciales M'_X con una textura que representa, en una escala de color, las distancias entre ambas mallas. Lógicamente en las zonas en las que no hay huecos se observará un color continuo equivalente al valor cero. Las variaciones de color aparecerán en las zonas correspondientes a los huecos. También es preciso puntualizar que es necesario utilizar una escala de color absoluta válida para la comparación entre todos los mapas de distancias, por lo que, habrá que calcular un máximo y un mínimo para todos los valores de distancias obtenidos en las $3j$ vistas parciales. Estos valores se incluyen en la Tabla 6.3 que aparece en el CAPITULO / ANEXO XXX. En éste también se muestran todos los mapas de distancias calculados tabulados en la Tabla 6.4, en función del algoritmo de relleno empleado: Roth, Criminisi y Software comercial.

Otra información que puede utilizarse para evaluar los resultados es la distancia media d_m obtenida para cada una de las $3j$ vistas parciales. Respecto a este parámetro, resulta interesante hacer una separación entre la distancia media global para una vista dada y las distancias medias para cada uno de los huecos rellenos en esa vista, definidas según las siguientes ecuaciones:

$$d_m = \frac{\sum_{i=1}^G (d(M_I, M'_X)_i)}{G} \quad (3.12)$$

$$d_{m_h} = \frac{\sum_{i=1}^{G_h} (d(M_I, M'_{Xh})_i)}{G_h} \quad (3.13)$$

siendo G el número de puntos que forma la nube de puntos g_n correspondiente a la malla de relleno M'_X . M'_{Xh} es una de las submallas que sirven de relleno del agujero h y G_h el número de puntos que la forman.

En las Tabla 3.2 y Tabla 4 pueden observarse los valores obtenidos para el cálculo de la distancia media por malla y la distancia media por huecos y malla, respectivamente, para cada uno de las j vistas parciales y en función de cada algoritmo de restauración de imágenes empleado. En dichas tablas anteriores se representa en rojo y negrita el valor de distancia más pequeño de los tres algoritmos de relleno, es decir, el mejor resultado tanto por hueco como por vista parcial completa.

Tabla 3.2: Distancia media por malla en función del algoritmo de image inpainting

Malla	DISTANCIA MEDIA POR MALLA (mm)		
	Algoritmo 1	Algoritmo 2	Algoritmo 3
<i>malla 1</i>	0,9347	1,5342	1,1498
<i>malla 2</i>	0,6284	0,8035	0,6564
<i>malla 3</i>	0,7742	4,8585	0,722
<i>malla 4</i>	0,4127	0,9938	0,4575
<i>malla 5</i>	0,64	0,9855	0,6949
<i>malla 6</i>	0,7153	2,2741	0,8207
<i>malla 7</i>	0,9653	1,698	0,8808
<i>malla 8</i>	0,7045	2,1069	0,8516
<i>malla 9</i>	0,8029	2,7879	1,0892
<i>malla 10</i>	0,6166	2,6012	0,7571
<i>malla 11</i>	0,7113	0,959	0,7526
<i>malla 12</i>	0,5521	2,6701	0,4256
<i>malla 13</i>	0,5159	0,8566	0,4512
<i>malla 14</i>	0,483	1,0492	0,5755
<i>malla 15</i>	0,3536	0,5818	0,3897
<i>malla 16</i>	0,5307	1,3913	0,5927
<i>malla 17</i>	0,8501	3,6414	0,6008
<i>malla 18</i>	0,4349	1,486	0,4885
<i>malla 19</i>	0,9262	2,0049	1,2902
<i>malla 20</i>	0,587	4,461	0,6276

Tabla 3.3: Distancia media por malla en función del algoritmo de image inpainting

Malla	Algoritmo de relleno	Distancia media por hueco (mm)						Distancia media Mallas (mm)
		Hueco 1	Hueco 2	Hueco 3	Hueco 4	Hueco 5	Hueco 6	
<i>ares</i>	1	1,2262	0,7529	0,8573	0,9281			0,9347
	2	2,2269	1,3113	1,1138	1,4716			1,5342
	3	1,458	1,1763	0,9095	0,9448			1,1498
<i>bebé</i>	1	0,4196	0,4051	0,7308	0,7983	0,3412	0,6529	0,4521
	2	0,5704	0,4941	0,6869	0,8458	0,6619	0,6126	0,6019
	3	0,6335	0,3963	0,8838	0,4351	0,5595	0,5247	0,5226
<i>bunny</i>	1	0,6389	0,6457	0,7324	0,9475			0,7742
	2	8,9338	6,6358	4,379	2,9937			4,8585
	3	1,3008	0,7846	0,0004	0,0004			0,722
<i>cerdo</i>	1	0,4683	0,3747	0,402	0,4233	0,3271		0,4127
	2	0,8663	1,59	1,4141	1,6301	0,5754		0,9938
	3	0,3678	0,1083	0,5435	0,4648	0,7465		0,4575
<i>cosa</i>	1	0,8153	0,4408	0,3502				0,64
	2	1,2448	0,7626	0,3391				0,9855
	3	0,3431	0,4022	0,9042				0,6949
<i>dinosaurio</i>	1	1,1524	0,619	0,6412				0,7153
	2	7,7129	1,4696	1,1281				2,2741
	3	0,7657	0,8603	0,7804				0,8207
<i>enano</i>	1	0,6633	0,8607	0,45	1,5533			0,9653
	2	1,7232	1,641	0,9106	2,2002			1,698
	3	0,6497	0,947	0,9855	0,6993			0,8808
<i>figura</i>	1	0,5701	0,6258	0,729	1,199	0,6454		0,7045
	2	2,3092	1,8261	2,4179	1,5772	2,2386		2,1069
	3	1,1336	1,3025	0,5801	0,7846	0,6852		0,8516
<i>leo</i>	1	0,7505	0,8267					0,8029
	2	2,776	2,7932					2,7879
	3	0,9793	1,2959					1,0892

Tabla 3.3: (continuación)

<i>marciano</i>	1	0,744	0,7055	0,477		0,6166
	2	1,395	3,8784	2,7064		2,6012
	3	0,5068	1,3566	0,4005		0,7571
<i>máscara</i>	1	0,529	0,7444	0,5595	0,789	0,7113
	2	0,9143	1,3707	1,0659	0,8113	0,959
	3	0,8618	0,7331	0,7224	0,6965	0,7526
<i>nenuco</i>	1	0,7209	0,6256	0,3034	0,4191	0,5521
	2	3,5419	2,5458	2,103	0,6706	2,6701
	3	0,2688	0,3481	0,4377	0,5309	0,4256
<i>patito</i>	1	0,3945	0,6463	0,3061		0,5159
	2	1,2165	0,6751	0,9206		0,8566
	3	0,4885	0,4802	0,3598		0,4512
<i>pato</i>	1	0,4577	0,7119	0,377		0,483
	2	0,9486	2,0427	0,5849		1,0492
	3	0,7713	0,5886	0,4577		0,5755
<i>peonza</i>	1	0,3795	0,3679	0,3077		0,3536
	2	1,0805	0,5272	0,5008		0,5818
	3	0,426	0,3927	0,2023		0,3897
<i>peque</i>	1	0,4039	0,5381	0,6484	0,5281	0,5307
	2	1,6651	0,7862	1,9116	0,6627	1,3913
	3	0,7146	0,4403	0,7208	0,4304	0,5927
<i>perrito</i>	1	0,6815	0,8764	1,0715	0,5783	0,8501
	2	3,841	2,1126	6,3709	0,6337	3,6414
	3	0,9152	0,1131	0,7648	0,5333	0,6008
<i>tetris</i>	1	0,2577	0,3277	0,4572	0,3328	0,4349
	2	0,3182	2,2414	0,5497	0,3902	1,486
	3	0,4591	0,3966	0,553	0,4292	0,4885
<i>trice</i>	1	0,9606	0,8021	1,0084		0,9262
	2	1,8191	2,3503	2,0009		2,0049
	3	0,9732	1,5952	1,3063		1,2902

Tabla 3.3: (continuación)

<i>vaca</i>	1	0,5296	0,6389	0,5629	0,587
	2	1,2174	9,5727	0,7208	4,461
	3	0,5006	0,7715	0,5364	0,6276

Comparativa de algoritmos por hueco

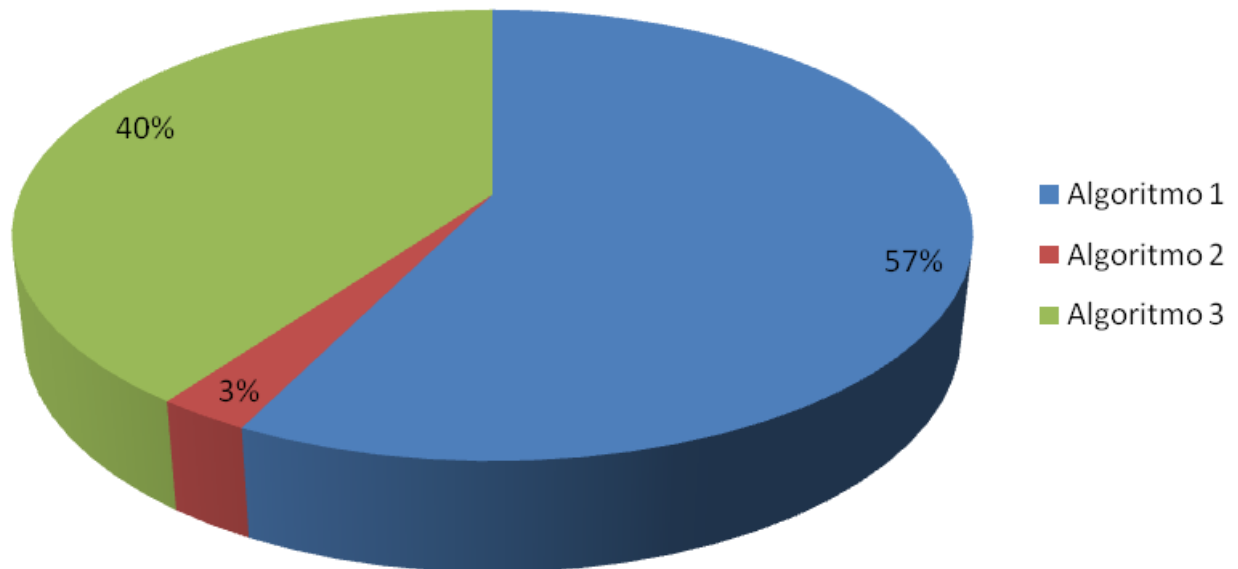


Figura 3.20: Comparativa de los algoritmos teniendo en cuenta la distancia media por hueco.

Una forma de resumir los resultados de forma gráfica es mediante la representación en gráfico de tarta, que se muestra en las figuras 3.20 y 3.21, en las que se puede ver claramente con qué algoritmo se obtienen mejores resultados por hueco y por malla completa (menor distancia media por hueco y malla):

- Con el algoritmo 1, el de Roth y Black, se consiguen las menores distancias medias por hueco en un 57% de los casos y menores distancias medias por malla en un 75% de los casos.
- Con el algoritmo 3, el software comercial de tratamiento de mallas y huecos, en un 40% de los casos analizados se obtienen menores distancias medias por hueco y en un 25%, menores distancias medias por malla completa.
- Y por último, con el algoritmo 2, el Criminisi y otros autores, sólo en el 3% de los casos se consiguen menores distancias medias por hueco. Sin embargo, si se tiene en cuenta la distancia media por malla completa, este algoritmo nunca obtiene menores distancias que los otros dos.

Por tanto, a la vista de estos resultados experimentales, el algoritmo más eficiente para el rellenado de huecos en mallas tridimensionales es el *Algoritmo 1*, que como se dijo, está basado en el método de restauración de [?].

A continuación se incluyen una serie de gráficos de barras en los que se puede comparar la distancia media por malla en función del algoritmo de rellenado (3.22) y la distancia media por hueco para cada una de las mallas analizadas (Tabla 3.4).

Comparativa de algoritmos por malla

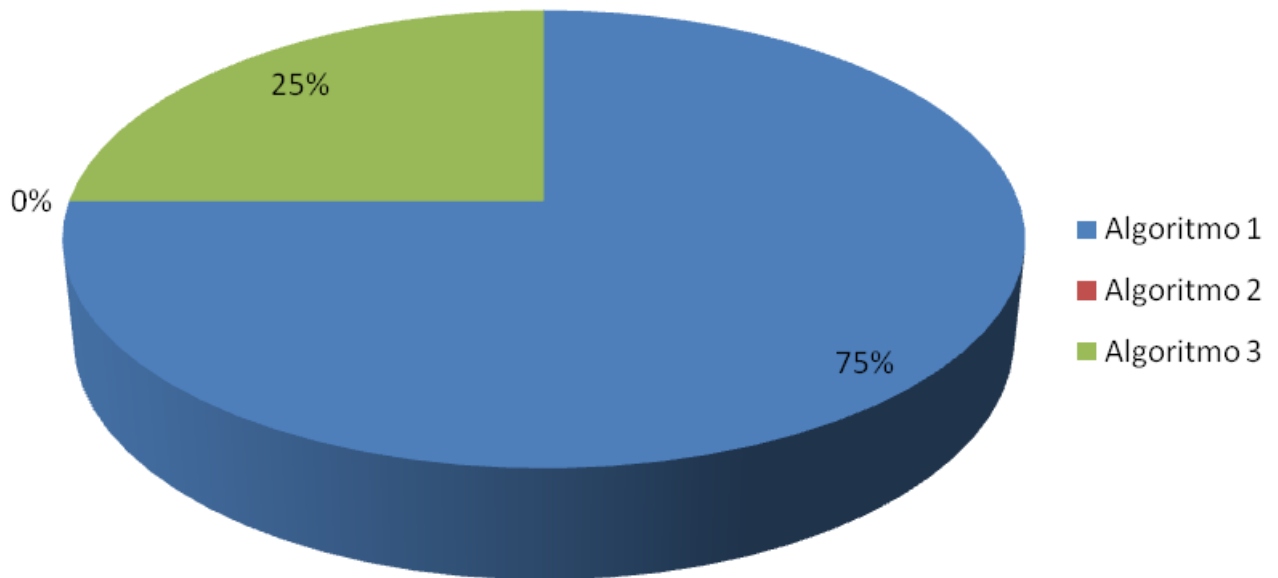


Figura 3.21: Comparativa de los algoritmos teniendo en cuenta la distancia media por malla.

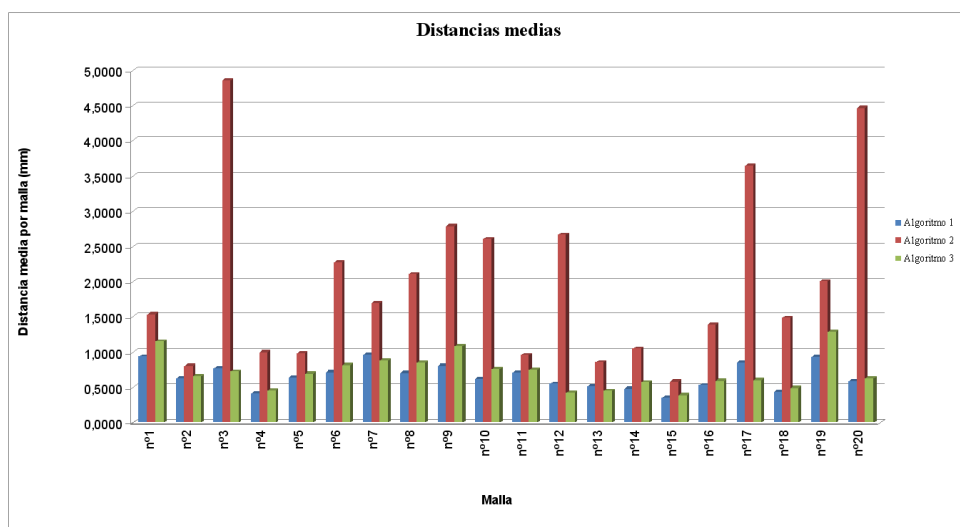


Figura 3.22: Representación de la distancia media por malla para los tres algoritmos.

Tabla 3.4: Representación de la distancia media por hueco en cada malla

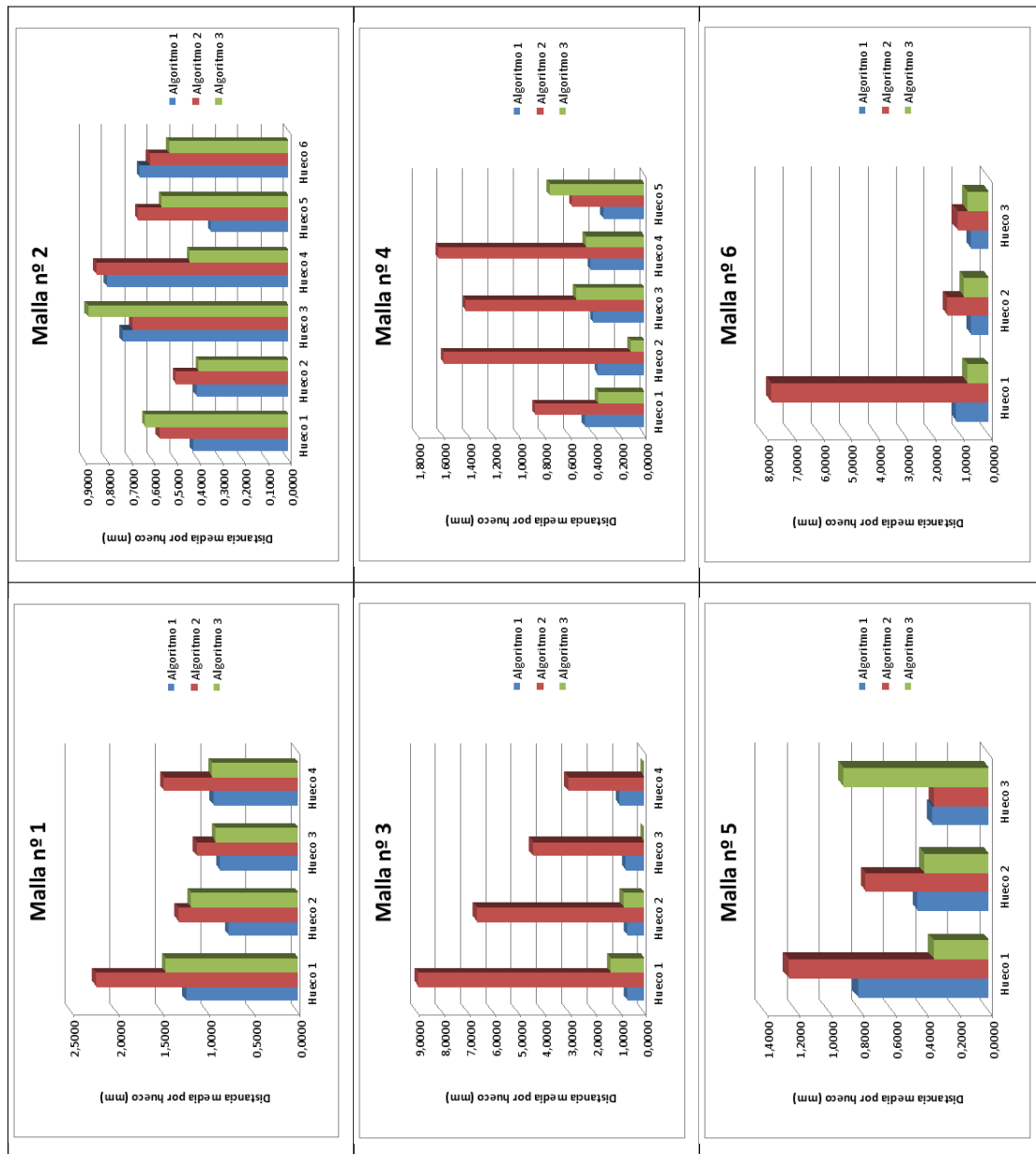


Tabla 3.4: (continuación)

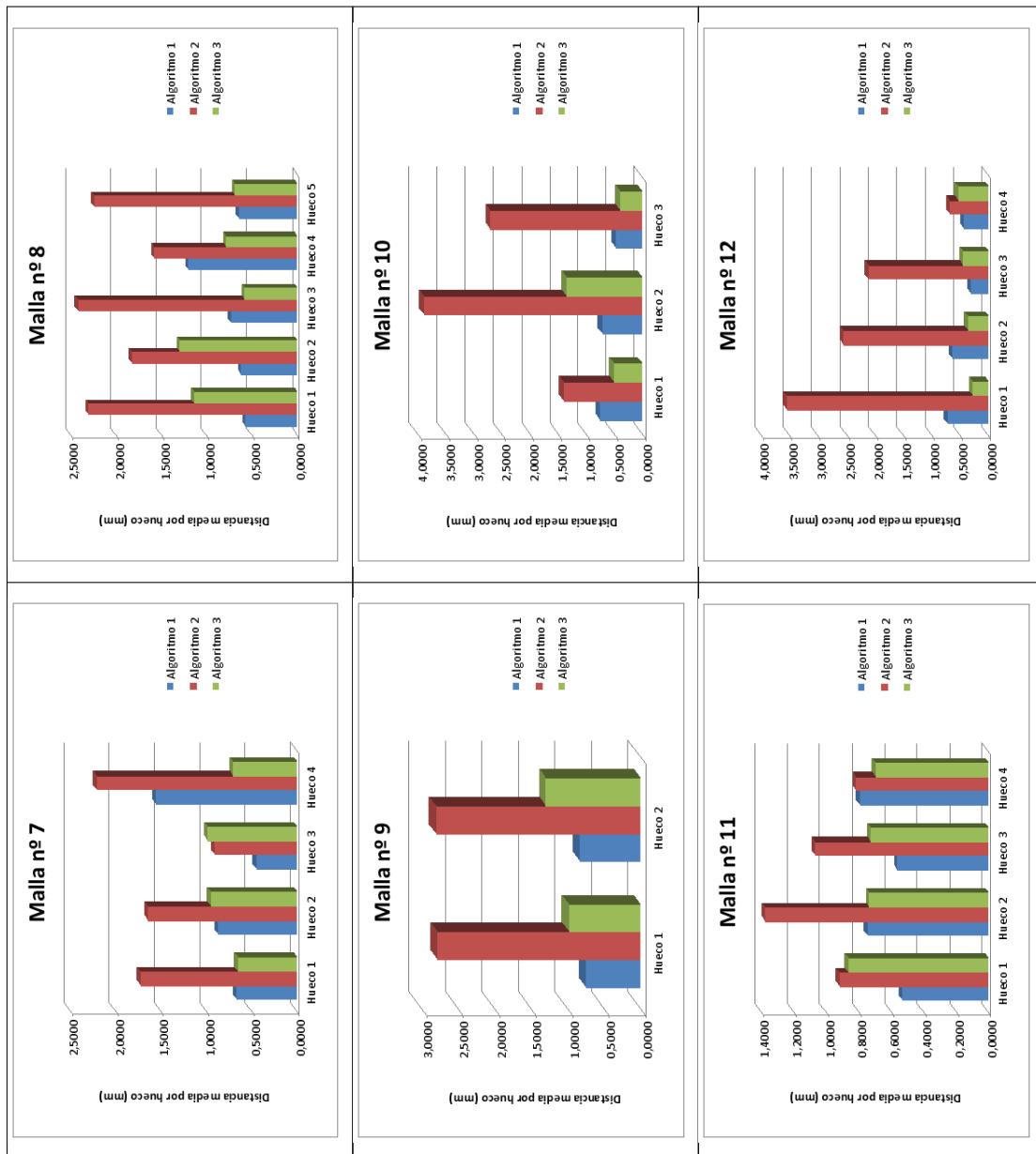


Tabla 3.4: (continuación)

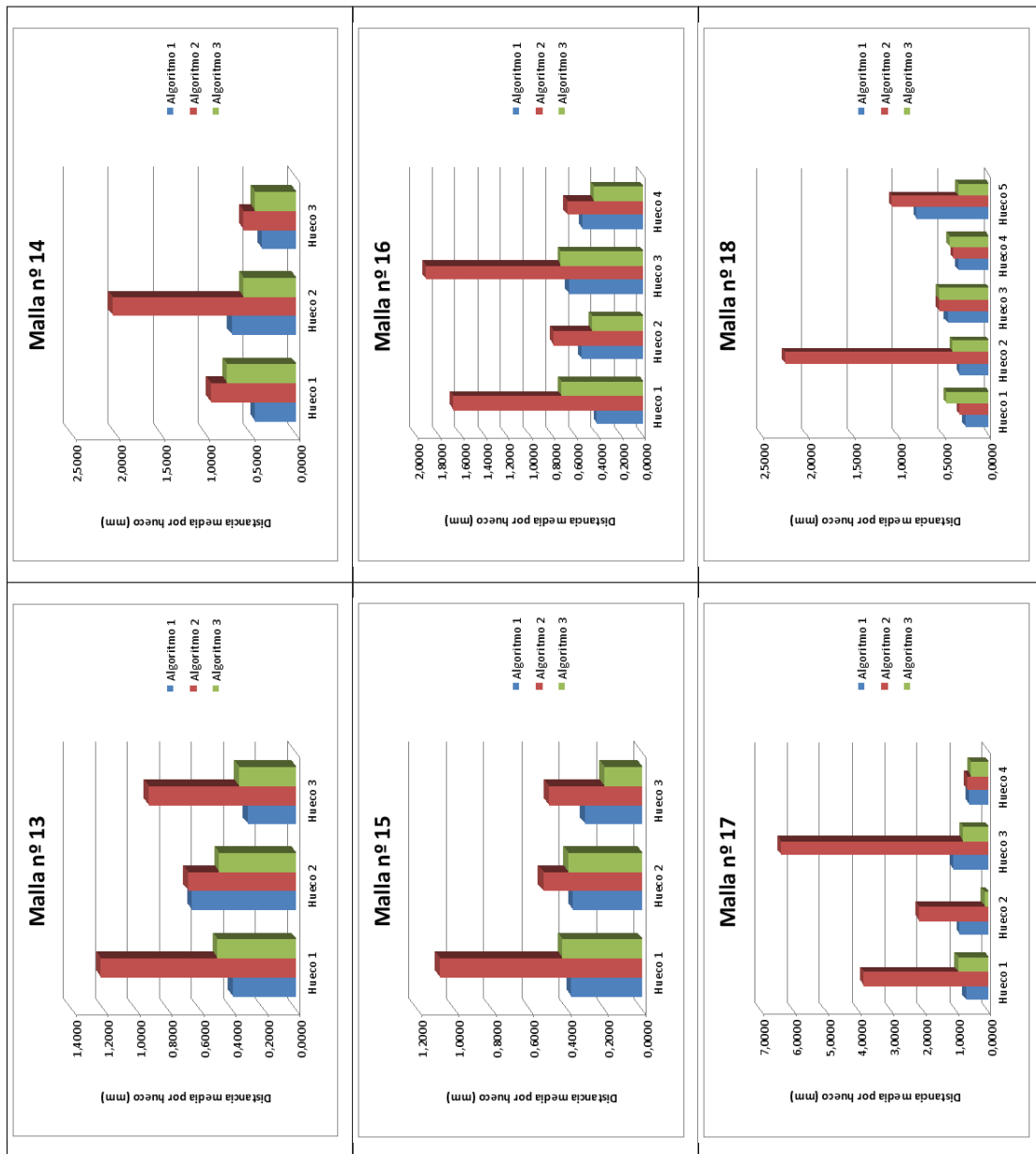
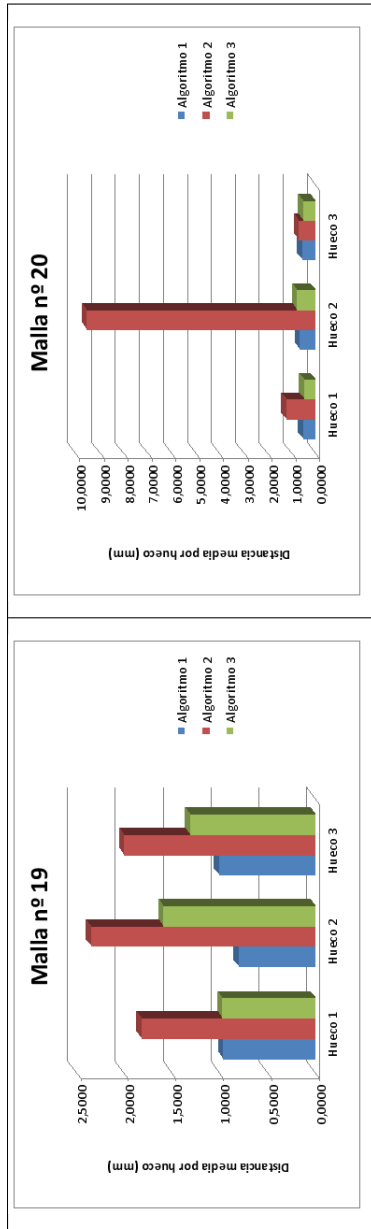


Tabla 3.4: (continuación)



En el ANEXO /CAPITULO pueden visualizarse las vistas parciales que han sido empleadas y los huecos que se les han generado en la Tabla 7.6. También pueden comprobarse los resultados de estos tres algoritmos sobre cada una de las vistas parciales empleadas en la Tabla 6.2. Por último, se representan gráficamente en las figuras 3.22 y 3.4, las distancias medias para cada vista y por cada hueco, respectivamente, obtenidas para las j vistas parciales utilizadas.

Por ejemplo la tabla ?? recoge algunas de las mallas reconstruidas.

A partir de las conclusiones obtenidas de las figuras 3.20 y 3.21, las mallas reconstruidas con el *Algoritmo 2* van a presentar una mayor desviación con respecto a la original en la mayoría de los huecos, con respecto a los *Algoritmos 1 y 3*. Por tanto, se decidió descartarlo como posible algoritmo de relleno antes de dar un paso más en la comparación de los resultados.

Seguidamente, y tras eliminar al *Algoritmo 2*, se estableció una nueva comparación entre las mallas restauradas mediante los otros dos *Algoritmos*, el que se basa en el método de Roth y los generados por el Software comercial (*1 y 3*). Así, para establecer un nuevo punto de vista en la comparación de resultados, se define un ratio de éxito como el ratio entre el número de huecos rellenados satisfactoriamente y el número total de huecos. La tasa que presenta el método propuesto era del 56 %, mientras que el software comercial presenta un 41,33 %.

Para obtener información completa y poder comprobar la robustez del algoritmo, se establece otra nueva comparación basada en las áreas de los agujeros. Para cada una de las mallas se ha determinado las áreas de los huecos y la razón entre este área y el área total de la malla. A esta lista de áreas y proporciones se le ha asociado una tercera columna con las distancias medias (errores) medidas en cada hueco. Seguidamente, se han ordenado estas listas de forma creciente con el área. De esta manera es más fácil averiguar qué método trabaja mejor, o lo que es lo mismo, tiene menor error, a medida que el área del agujero es mayor. Para un hueco dado, si un error de uno de los algoritmos es más pequeño que el otro, se incrementa un contador de éxitos. Por lo tanto, se genera una lista acumulativa. La figura 3.23 (a) representa la tasa de éxito con respecto al número total de huecos, frente a las áreas de los agujeros. A la vista de esta gráfica, es inmediato concluir que el método propuesto produce mejores resultados que el comercial.

Además, en lugar de las áreas absolutas de los agujeros, se ha calculado el porcentaje del area para cada hueco en relación al área total de la malla. De esta manera, se ha establecido una nueva confirmación para comprobar la validez de la conclusión anterior. Como puede verse en la figura 3.23 (b), el método de relleno que se pretende validar ofrece una ventaja significativa con respecto al software comercial, teniendo en cuenta este parámetro.

Lo que se puede deducir, al observar la figura 3.23, es que hay un valor particular del área del agujero a partir del cual la tasa de éxito del software comercial nunca supera al método propuesto. Esto da una idea de la robustez del algoritmo propuesto para el relleno de huecos.

Para concluir, decir que el método desarrollado ha sido aplicado también en varias mallas de esculturas de la colección del Museo Nacional de Arte Romano de Mérida, que han sido digitalizadas en el transcurso de los trabajos de investigación. En la figura 3.24 pueden verse algunos ejemplos de estas mallas.

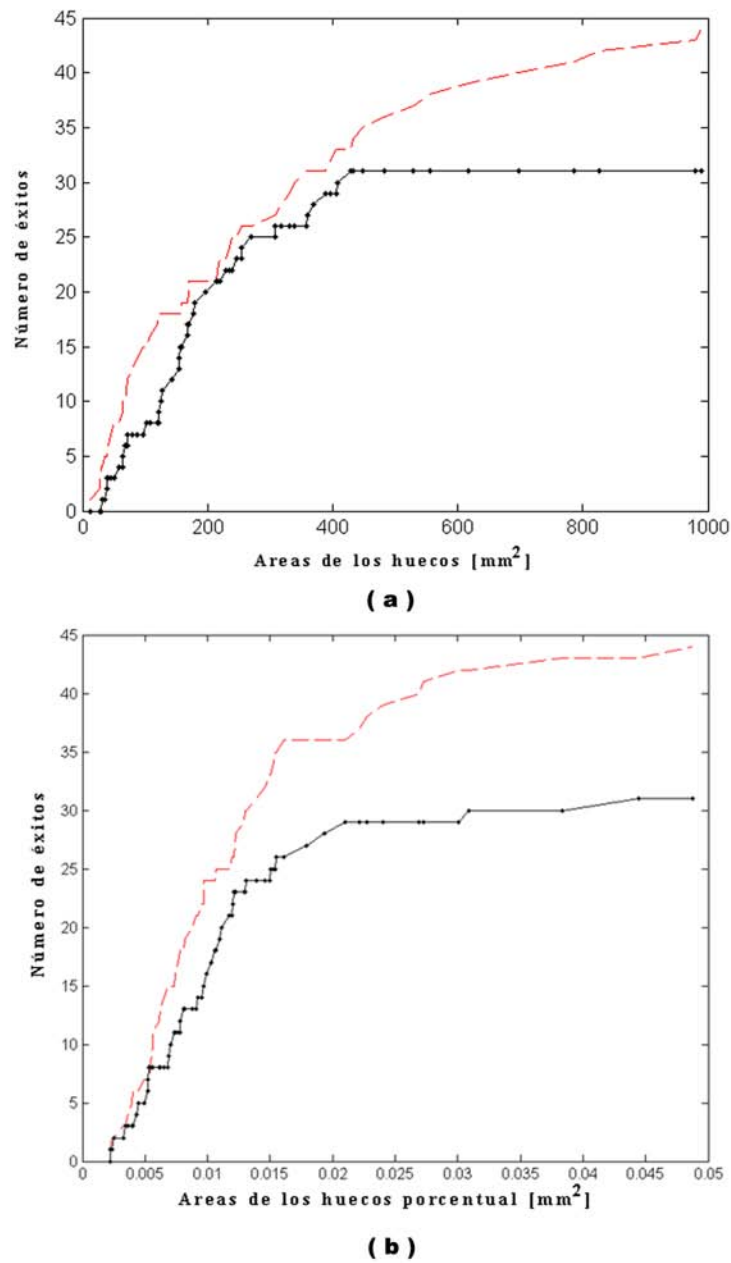


Figura 3.23: Representación de la tasa de éxito frente a (a) el área total de los agujeros y (b) el área porcentual. La línea continua es para el método propuesto, y la línea de puntos para el algoritmo comercial.

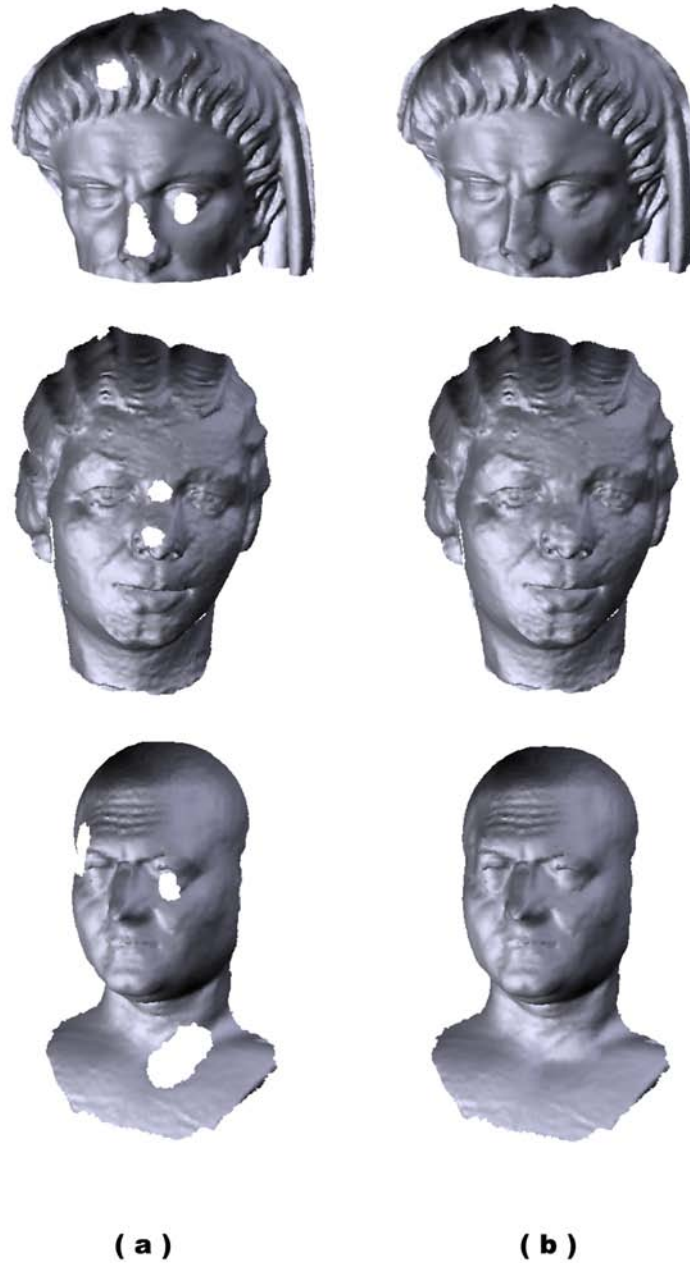


Figura 3.24: Mallas (a) originales y (b) restauradas, digitalizadas de la colección del Museo Nacional de Arte Romano de Mérida.

3.9. Resumen

En este capítulo se ha presentado un método para el rellenado de huecos en mallas 3D. Para esto es necesario, en primer lugar, generar una imagen 2D como entrada del algoritmo de restauración de imágenes. Debido a la influencia de la resolución de la rejilla (pixels) en la pérdida de datos cuando los datos 3D son proyectados, se ha determinado la resolución óptima para minimizar esa pérdida.

Para probar la efectividad del método se ha aplicado sobre vistas parciales a las que se le han generado huecos artificiales. Se han establecido dos comparaciones para estudiar las diferencias entre los diferentes métodos: la distancia entre las mallas originales y reparadas (por malla completa y por zona reparada) y la efectividad con respecto al área del hueco. Dos conclusiones importantes se pueden obtener. La primera es que el método propuesto ofrece errores más pequeños al utilizar el algoritmo de Roth, que en los otros casos. La segunda es que se ha probado que este es también más adecuada para rellenar huecos de cualquier tamaño. Por tanto, se ha demostrado que se trata de un método de rellenado de huecos robusto.

3.10. Resultados experimentales

En este capítulo se incluyen los resultados obtenidos aplicando los tres algoritmos de image inpainting para el rellenado de huecos en veinte mallas tridimensionales. Estos resultados se refieren a la distancia media por cada uno de los huecos artificiales de cada malla y a la distancia media por malla, que se obtienen comparando la malla original con la malla reconstruida de la forma que se ha explicado en la sección anterior.

Capítulo 4

Definición de un método general para el rellenado de huecos en mallas 3D

4.1. Introducción

En el capítulo anterior se proponía la validación del método desarrollado mediante su aplicación en vistas parciales de baja resolución y a las que se les habían generado huecos artificiales. Tras la observación de los resultados y su comparación con otros métodos de relleno, se concluyó que la técnica propuesta ofrecía buenos resultados, en huecos de diversos tamaños, formas y curvaturas.

Como ya se mencionó en el capítulo 2, el problema de los huecos se presentaba durante la reconstrucción de modelos 3D de objetos completos. Se decía también que es un problema que no tiene una solución trivial y que sea genérica para cualquier tipo de hueco y superficie que se pueda presentar. Es por ello que se plantea en este capítulo, la extensión del método propuesto a la resolución del problema en objetos completos. Estos objetos presentan huecos que se han generado durante las distintas fases de reconstrucción de los mismos, tal y como se detallaba en el capítulo 2.

A pesar de mantener las premisas del método que se desarrolló para su aplicación en vistas parciales, las particularidades del problema en objetos completos, hace que se tengan que replantear las etapas del proceso. De esta forma, en el capítulo que nos ocupa se estudiará la nueva situación y se planteará la definición de un método general para el rellenado de huecos en mallas de objetos completos.

A continuación se generaran una serie de resultados experimentales sobre mallas obtenidas tanto de la digitalización mediante sensores de rango, como de repositorios de mallas disponibles en Internet. Seguidamente se establecerán unos métodos de medida que sirvan para analizar los resultados que se generen.

Por tanto, la división del capítulo se hará como sigue: en primer lugar, en la sección 4.2, se presentará el esquema de funcionamiento básico del método, enumerando las etapas que lo componen. En las secciones sucesivas, desde la sección a la, se detallarán cada una de las etapas. Seguidamente se presentarán algunos de los resultados experimentales obtenidos y, finalmente se expondrán las conclusiones.

4.2. Descripción general del método para mallas completas

En comparación con la situación descrita en el capítulo anterior, la idea de base del método es la misma para ambos tipos de entrada del proceso para vistas parciales y para mallas completas. Pero en las últimas, se añaden más factores a tener en cuenta en el problema a resolver. A lo largo del capítulo anterior se han ido adelantando algunas diferencias entre ambos tipos durante el estudio de las diversas etapas. En este capítulo se va a replantear el método de relleno presentado para vistas parciales, añadiéndole las pertinentes modificaciones para que sea aplicable en objetos completos y estudiando las peculiaridades del nuevo tipo de entrada del proceso.

Por tanto, el método de relleno aplicado a objetos completos, puede ser esquematizado en los siguientes pasos (representados en la figura 4.1):

1. Identificación de los $h_i (i = 1, 2, \dots, k, \dots, m)$ huecos en la malla total M_T .
2. Selección del hueco h_k .
3. Selección de la porción de malla $M_{v_i} \subset M_T$, que es la porción de malla que se encuentra en el entorno del hueco h_i .
4. Transformación 3D a 2D para obtener la imagen de rango, I_{h_i} , asociada a la malla con huecos $M_{v_i} \subset M_T$.
5. Aplicación del algoritmo de *image inpainting*. La imagen reconstruida se denotará por I_{f_i} .
6. Transformación 2D a 3D y fusión de los datos de relleno con la malla con agujeros. 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 .
7. Repetir el proceso desde 2, para el siguiente valor de i . Ahora se usará M'_T .

Se observa a primera vista que, con respecto a la aplicación del método en vistas parciales, se han incluido tres nuevas etapas (sección 3.3). La primera, la identificación de huecos, también se llevaba a cabo en la situación anterior, aunque, ya que se generaban artificialmente sobre vistas parciales, los huecos a rellenar no presentaban una gran complejidad y su identificación, dada la regularidad de las vistas parciales, era relativamente sencilla. Cuando se trata de aplicar la técnica en objetos completos reales, los huecos se presentan en un lugar y con unas características desconocidas, por lo que se ha trasladado a este capítulo el tratamiento del proceso de identificación de huecos.

Otra de las etapas que se añaden es la que ocupa el segundo lugar. Dado que los datos de entrada van a ser objetos completos, no puede hacerse una transformación a 2D de todo los puntos que los forman. Por lo que habrá que extraer una porción de la malla adecuada para realizar dicha transformación y obtener una imagen de rango para cada hueco.

Por último, también se ha incluido como nueva etapa, aunque no conlleva en sí un proceso, la enumerada en último lugar. Con esta etapa se hace referencia a otra diferencia que surge al utilizar objetos completos. En el capítulo anterior, al tratar con vistas parciales, se realizaba una única vez la secuencia de etapas que se proponía. Es decir, se generaba una única imagen

Figura 4.1: Etapas del método de relleno de huecos desarrollado

de rango de toda la vista parcial, sobre la que se iban rellenando los diferentes huecos. Al final, se aplicaba la transformación inversa a 3D, obteniendo una única vista parcial reconstruida, con los huecos rellenos. En el caso que nos ocupa, no es posible una única ejecución del algoritmo dado que, como ya se ha mencionado, es imposible obtener una única proyección o imagen de rango de todo el objeto. Así, habrá que ir repitiendo el proceso para cada una de las porciones de malla extraídas para el relleno de cada hueco.

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, que son impuestas por el propio funcionamiento del método. Estas exigencias son las siguientes:

- La malla ha de ser una variedad bidimensional: cada arista es compartida como máximo entre dos triángulos. Esta es una de las exigencias habituales de la mayoría de algoritmos que trabajen con mallas triangulares, ya que se trata de un error en la representación de la malla que hay que evitar.
- El tamaño de los triángulos ha de ser aproximadamente constante para toda la malla. Para obtener la imagen de rango, la malla se proyecta sobre un plano o rejilla de proyección la cual no es más que una cuadrícula totalmente regular, por lo que interesa que la proyección de los puntos se distribuya de forma regular sobre la cuadrícula para evitar las pérdidas que se comentarán en la sección 4.5.

A pesar de estas exigencias, no se restringe en ningún momento el ámbito de aplicación del método de relleno, ni le restan generalidad, ya que cualquier malla puede ser procesada, antes de aplicarle el método de relleno, para que cumpla con tales exigencias. Este procesado previo no significa una modificación significativa de la superficie representada por la malla.

A continuación se explicarán detalladamente cada una de las etapas de la técnica de relleno que se ha propuesto.

4.3. Identificación de los huecos

La primera etapa del método de relleno es la búsqueda de los huecos que han de ser rellenados y, más concretamente, la determinación de los bordes de esos huecos. Antes de adentrarse en el algoritmo de identificación de huecos, es necesario recordar en qué se basaba el método de representación de objetos 3D mediante malla triangular.

En la sección 2.3 anterior del capítulo 2, se explicaban diversas características de las mallas triangulares. Concretamente, se detallaban las propiedades que cumplían las variedades bidimensionales cerradas, que son el tipo de mallas que se pretenden obtener al aplicar el algoritmo de relleno de huecos.

El problema es que, la particularidad que presentan las mallas a la entrada del algoritmo es que tienen huecos. Este hecho se traduce en que ya no son de aplicación todas las expresiones que aparecían en dicha sección, o al menos en toda la extensión de la malla. Concretamente, con la aparición de los huecos ya no se cumplen las expresiones (i) y (ii), las cuales habría que cambiarlas por:

i') Conectividad de los vértices: todos los vértices de la malla están conectados con al menos otros dos vértices vecinos, o lo que es lo mismo, tienen al menos dos aristas incidentes:

$$\forall v_i \in \mathcal{V}, \exists \mathcal{A}_S \subset \mathcal{A} / \forall a_h \in \mathcal{A}_S, v_i \in a_h \wedge \text{card}(\mathcal{A}_S) \geq 2$$

ii') Pertenencia de aristas a triángulos: dado que existen hueco siempre existirá un subconjunto de aristas que sólo pertezcan a un único triángulo:

$$\exists \mathcal{A}_S \subset \mathcal{A} / \forall a_h \in \mathcal{A}_S \wedge \forall t_i \in (\mathcal{T} - \mathcal{T}_S), t_i \cap t_j \in \{\emptyset\} \iff a_h \subset \mathcal{T}_S \wedge t_j \in \mathcal{T}_S$$

La razón de este cambio es porque en las zonas de los huecos ha habido una 'rotura' de esa red de triángulos con lo cual ahora existirán aristas, las del borde de los huecos, que sólo pertenecen a uno de ellos, lo que también dará lugar a vértices con un grado 2.

De esta manera, para identificar los huecos en una malla se procederá como sigue. Dada una malla triangular con huecos, M_T , se tienen los siguientes elementos que la definen:

- Nube de puntos que la forman $\mathcal{P} = \{p_1, \dots, p_n\}$ con $p_i \in \mathbb{R}^3$.
- 3-Tupla de entidades $\{\mathcal{V}, \mathcal{A}, \mathcal{T}\}$, siendo \mathcal{V} una lista de índices $\{v_1, \dots, v_n\}$ que hace referencia a elementos de la nube de puntos \mathcal{P} ; \mathcal{A} es la lista de aristas $\{a_1, \dots, a_k\}$ en la que cada elemento a_i se representa por una pareja de índices $\langle u, w \rangle$ que indican las posiciones que ocupan en \mathcal{V} los vértices que la forman; \mathcal{T} es la lista de triángulos o caras que componen la superficie, $\{t_1, \dots, t_m\}$, en la que cada elemento t_i se representa por un grupo de tres índices $\langle r, s, t \rangle$, donde r, s y t son índices que indican la posición en que se encuentran en \mathcal{V} los vértices que lo componen.

Para facilitar la representación se ordena la lista de puntos \mathcal{P} de forma que los índices de \mathcal{V} sea una secuencia ordenada que vaya desde 1 hasta el número total de vértices, n . Por otra parte, resulta evidente que existe redundancia en la información asociada a la malla, ya que a partir de la lista de aristas \mathcal{A} puede construirse fácilmente la de triángulos \mathcal{T} y viceversa.

A continuación, habrá que buscar las aristas de la malla que definen los bordes de los huecos, es decir, las que sólomente sean adyacentes a un único triángulo. Para ello, se seguirán las siguientes etapas:

1. Se aplica un criterio de ordenación en los triángulos, de tal forma que se genera \mathcal{T}' , en donde cada elemento $\langle r, s, t \rangle$ se ordena de forma creciente, donde $r < s < t$.
2. A partir de \mathcal{T}' , se extrae la lista de aristas \mathcal{A}' de tal manera que cada elemento de \mathcal{T}' , $\langle r, s, t \rangle$, da lugar a tres elementos, $\langle r, s \rangle$, $\langle s, t \rangle$ y $\langle r, t \rangle$. Con esto resulta una lista de aristas en la que, a diferencia de \mathcal{A} , existirá repetición de elementos.

3. Para facilitar la búsqueda, se aplica la siguiente función de hash a \mathcal{A}' :

$$\varphi = (\eta + \theta) / \vartheta \quad (4.1)$$

donde, si se representa cada arista por $\langle u, w \rangle$, $\eta = w - u$, $\theta = u \cdot w$ y $\vartheta = (u + w) / \sqrt{w}$. Por tanto, tras aplicar esa función se obtiene una nueva lista \mathcal{A}'_φ , en la que ahora cada arista viene definida por su parámetro φ .

4. Se construye un vector c_φ de la misma dimensión que \mathcal{A}'_φ y que almacena en cada posición $c_\varphi(j)$, el número de veces que está repetido el elemento $\mathcal{A}'_\varphi(j)$ en \mathcal{A}'_φ .
5. Si se toman los elementos de c_φ que valgan 1, se tendrán las aristas que sólo aparecen una vez en la lista, o lo que es lo mismo, las que sólo están compartidas por un único triángulo. A partir de ahí, se construye la lista de aristas que son bordes de huecos, \mathcal{A}_b , que se correspondería con:

$$\mathcal{A}_b = \mathcal{A}'_\varphi(k) / c_\varphi(k) = 1 \quad (4.2)$$

6. Se procede a la búsqueda de los posibles caminos cerrados que se forman con las aristas contenidas en \mathcal{A}_b . Para tal fin, se comienza creando el primer borde de hueco seleccionando una primera arista de \mathcal{A}_b :

$$B_i = \mathcal{A}_b(1) = \{ \langle v_1, v_2 \rangle \} \quad (4.3)$$

7. Se elimina la arista de \mathcal{A}_b la arista que se ha añadido a B_i . Se toma v_1 como primer vértice del borde de hueco.
8. Seguidamente se busca en \mathcal{A}_b , la arista en donde aparece v_2 , la cual es eliminada de \mathcal{A}_b , y añadida a B_1 , quedando:

$$B_i = \{ \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle \} \quad (4.4)$$

9. Volver a 8 pero ahora buscando v_3 y así sucesivamente hasta que se obtenga una arista que contenga también al primer vértice del borde v_1 . En tal situación, se habrá obtenido el primer borde de hueco B_i .
10. Si aun quedan elementos en \mathcal{A}_b , se continúan extrayendo de la misma forma más bordes de hueco, hasta vaciar la lista \mathcal{A}_b . De esta forma se extraerán los bordes de hueco B_n , que tendrán la forma:

$$B_n = \{ \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \dots, \langle v_k, v_{k+1} \rangle, \dots, \langle v_{P-1}, v_P \rangle, \langle v_P, v_1 \rangle \} \quad (4.5)$$

en donde cada borde tendrá un número P de aristas, diferente para cada hueco.

11. A partir de los distintos B_n se define el conjunto de bordes de huecos de la malla como:

$$B_c = \{ B_1, B_2, \dots, B_n \} \quad (4.6)$$

Una vez determinado este conjunto, se procederá al relleno de los huecos, de en 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 4.2 puede observarse un ejemplo de identificación de tres huecos en una malla.

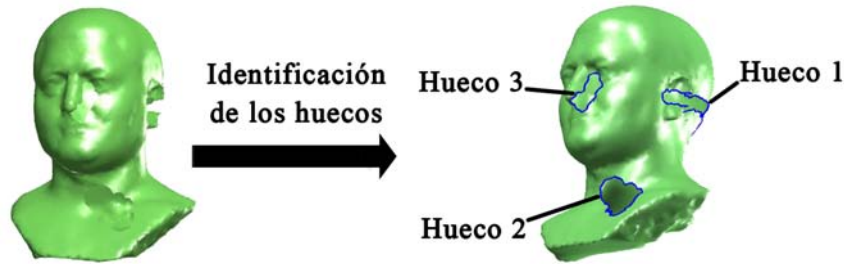


Figura 4.2: Etapa inicial de identificación de los h_i ($i = 1, \dots, m$) huecos en la malla total M_T .

Por consiguiente, una vez finalizada esta etapa de identificación de huecos, se elegirá un $B_k \subset B_c$ y se continuará con la siguiente etapa del proceso, la selección de la porción de malla, que será explicada a continuación.

4.4. Selección de la porción de malla

Como ya se ha mencionado, para proceder al relleno de un hueco es necesario obtener una imagen de rango de la zona en donde aparezca el hueco. La extensión de esta zona tendrá que ser lo suficientemente grande con respecto al tamaño del hueco para que la información que aporte sea representativa para que el algoritmo de restauración opere de forma correcta. La elección de la extensión de esa porción de malla que resulte más adecuada será abordada posteriormente. En primer lugar, se va a describir cómo se realiza la extracción de dicha porción.

Una vez identificados los bordes de los huecos B_c en la etapa anterior, se selecciona uno de los bordes B_k . A partir de ahí se procede según las siguientes etapas:

- Si el borde B_k está compuesto por un conjunto de g aristas \mathcal{A}_{B_k} , de ahí puede extraerse el conjunto de g vértices v_{B_k} que se forma como sigue:

$$B_k = \{ \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \dots, \langle v_{g-1}, v_g \rangle, \langle v_g, v_1 \rangle \}; v_{B_k} = \{v_1, v_2, v_3, \dots, v_g\} \quad (4.7)$$

- A partir de v_{B_k} y \mathcal{T} , puede extraerse un subconjunto de triángulos que contengan uno o dos vértices de v_{B_k} . Este subconjunto conforma un anillo de triángulos R_1 del que se dice que tiene grado de vecindad uno con B_k ya que los vértices que lo forman tienen, como máximo, un grado de vecindad uno con los vértices v_{B_k} . Este anillo R_1 consiste en una porción de malla con dos bordes, uno interior que coincide con B_k y otro exterior B_{e1} . De igual forma, puede calcularse el anillo R_2 utilizando este último borde exterior B_{e1} , y así, obtenerse una porción de malla con borde interior B_{e1} y borde exterior B_{e2} . En este caso, se dice que R_2 tiene un grado de vecindad dos con B_k . De forma general, aplicando el proceso v veces se obtendría el anillo R_v con borde interior $B_{e(v-1)}$ y borde exterior B_{ev} y con grado de vecindad v , pues sus vértices tienen, como máximo, un grado de vecindad v con los vértices v_{B_k} . De esta manera, a partir de todos esos anillos, se extrae la porción de malla M_v correspondiente al borde de hueco B_k , que puede expresarse como:

$$M_v = \bigcup_{i=1}^v R_i \quad (4.8)$$

Diremos que esta porción de malla tiene un grado de vecindad v con respecto a B_k que coincide con el mayor grado de vecindad de los anillos que la componen. M_v 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. Así, si 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 . En la figura 4.3 se observa el proceso de selección de una porción de malla con grado de vecindad $v = 6$. En la figura 4.4 se representan las diferentes porciones de malla extraídas para cada uno de los huecos mostrados en la figura 4.2.

4.5. 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 , de la porción de malla que se ha seleccionado. Al igual que en el capítulo anterior, se hace necesaria la búsqueda de un plano de referencia óptimo sobre el que proyectar los nodos de la malla, proyección que generará la imagen de rango. Para la obtención del plano de referencia se habrán de seguir los mismos pasos que en el capítulo anterior, con el fin de fijar el valor de las siguientes parámetros:

1. Plano de proyección sobre el que proyectar.
2. Resolución de la imagen de rango.

Cuando se trabajaba con vistas parciales, dado que eran obtenidas directamente mediante digitalización del objeto con un escaner, se decía que la vista parcial venía definida en un sistema de referencia ligado al escaner, con el plano XY coincidente con el plano imagen del sensor y el eje Z perpendicular a dicho plano imagen. Así, la orientación del plano de referencia se escogía paralela al plano XY de referencia. Esto facilitaba enormemente la generación de la imagen de rango ya que reducía el problema al cálculo de la resolución de la imagen de rango óptima.

En el caso que nos ocupa, se añade la complicación de tener que buscar un plano de referencia adecuado para cada uno de los huecos, del mismo objeto, que se desee rellenar, como se observa en la figura 4.5. A continuación se detallará cómo se determinan cada uno de los parámetros mencionados anteriormente: plano de proyección óptimo y resolución de la imagen de rango.

4.5.1. Plano de proyección sobre el que proyectar

En este punto lo que se persigue es la determinación de la orientación, posición y tamaño que ha de tener el plano de proyección con respecto a la superficie 3D. Es evidente que el plano de proyección que se elija tendrá influencia en la capacidad de representación que tenga la imagen con respecto a los datos 3D. En la figura 4.6 (a) se puede observar una elección errónea del plano de proyección ya que coinciden las proyecciones de dos puntos 3D en un mismo punto de plano, lo que da lugar a una función bidimensional multivaluada. Sin embargo, en la figura 4.6 (b) el plano de proyección elegido no da lugar a situaciones como la anterior.

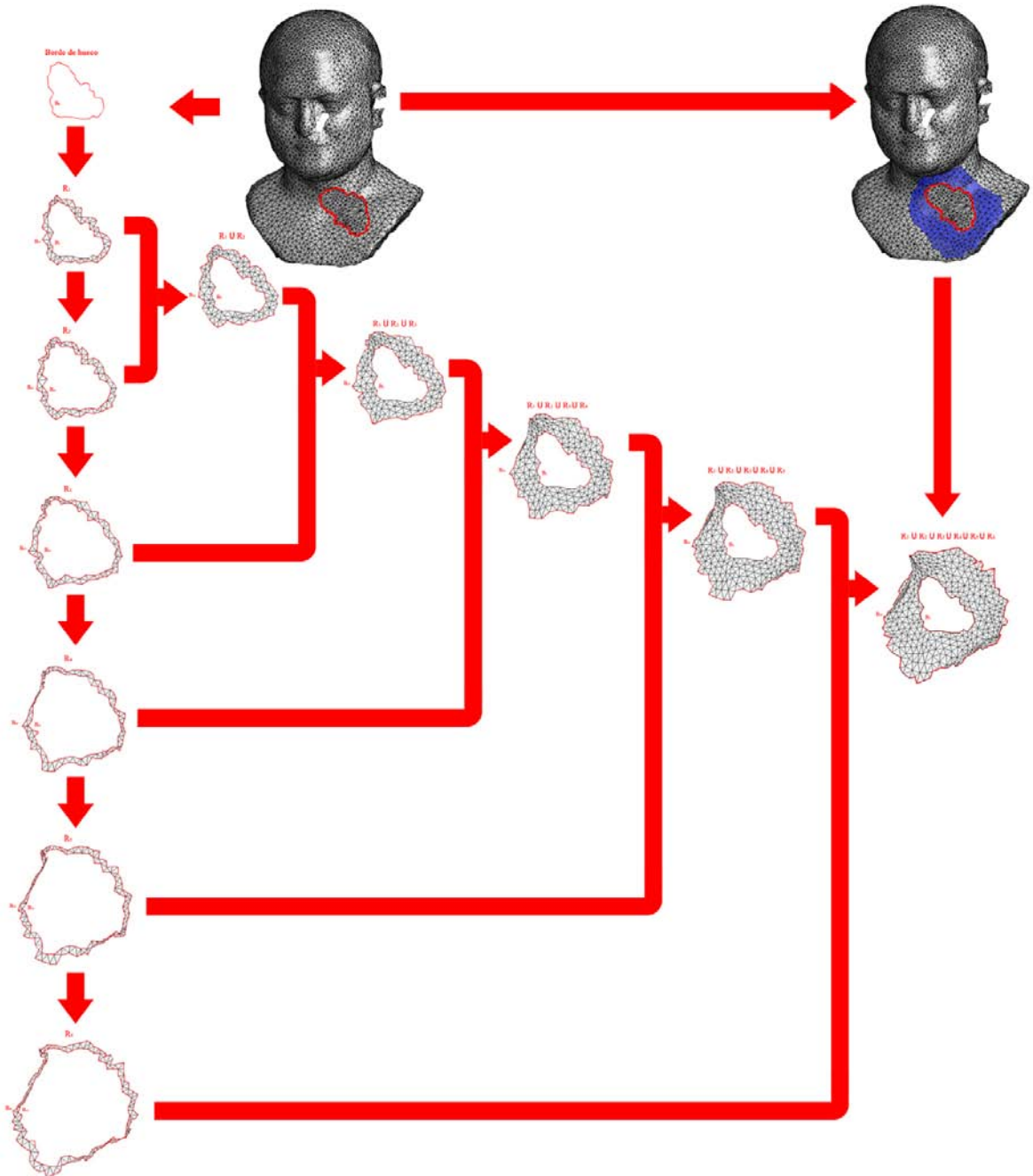


Figura 4.3: Procedimiento de selección de la porción de malla mediante la obtención de los anillos R_i , a partir del borde de hueco B_k

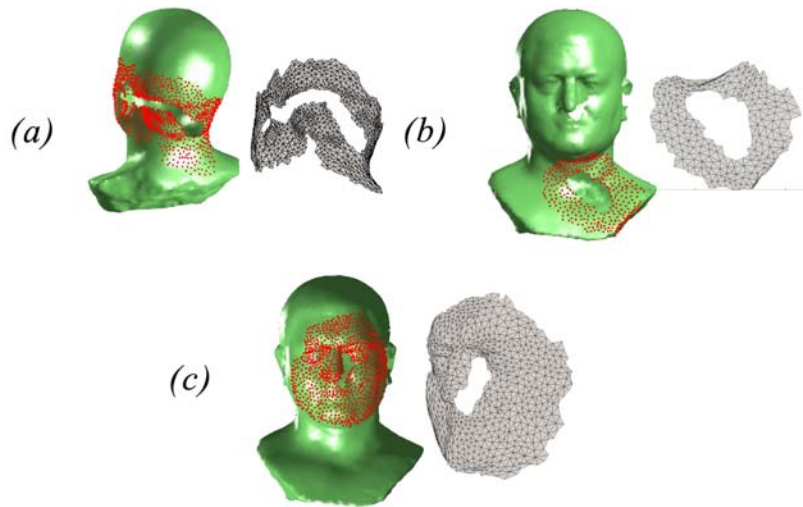


Figura 4.4: Selección de las porciones de malla para cada uno de los huecos que aparecen en la figura 1: (a) Hueco 1; (b) Hueco 2; (c) Hueco 3. Se han utilizado distintos valores de v en cada caso: $v = 7$; $v = 5$; $v = 11$, respectivamente.

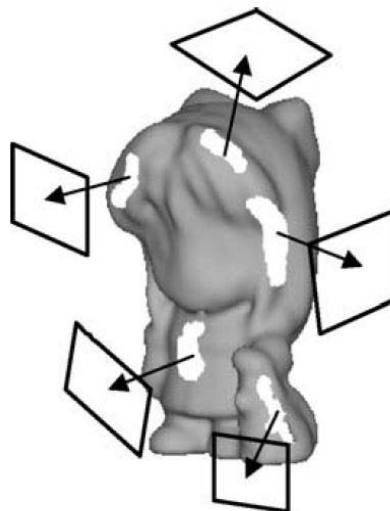


Figura 4.5: Selección de diferentes planos de referencia o de proyección para cada uno de los huecos que se van a rellenar.

Figura 4.6: Elección del plano de proyección para hacer la transformación 3D a 2D: (a) elección errónea; (b) elección idónea

[Elección del plano de proyección para hacer la transformación 3D a 2D: (a) elección errónea; (b) elección idónea.]

En el proceso de generación de la imagen de rango, la región de mayor interés es el propio hueco que ha de ser rellenado, ya que será el elemento de la imagen sobre el que actúa el algoritmo de restauración de imágenes. Por ello, se tendrá que elegir una orientación según la cual el hueco, tras efectuar la proyección sobre el plano, quede representado de la mejor forma posible. Pueden establecerse diversos criterios para la elección idónea de esa orientación, y de entre ellos se han elegido tres, por su buen funcionamiento en el proceso de relleno. Estos criterios son:

- Orientación basada en la asociación de normales al hueco.

Según este criterio, el plano de referencia estará definido por un vector normal que es paralelo al vector asociado a los triángulos del borde de hueco. Así, para calcularlo, dado un borde de hueco B_k , se buscan los q triángulos que componen el anillo R_1 , con grado de vecindad 1, que son los que se muestran en la figura 4.8(a). Cada uno de estos triángulos tendrá asociado un vector normal unitario \vec{n}_i . El vector normal asociado al borde de hueco, \vec{N}_k , o lo que es lo mismo, la normal que se utilizará para determinar el plano de proyección, puede obtenerse como sigue:

$$\vec{N}_S = \sum_{j=1}^q \vec{n}_j \quad (4.9)$$

$$\vec{N}_k = \frac{\vec{N}_S}{\|\vec{N}_S\|} \quad (4.10)$$

es decir, se calcula como la suma normalizada de los vectores normales a los triángulos que componen R_1 (figura 4.8(b)).

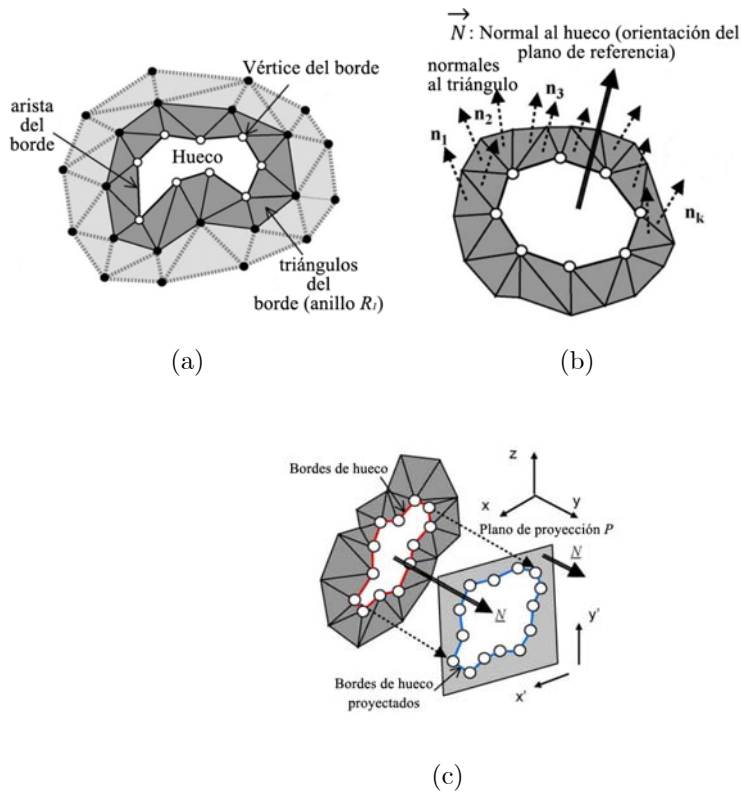


Figura 4.7: (a), Anillo de triángulos que pertenecen al borde del hueco; (b) Normal al hueco obtenida como la media de las normales a las caras que forman el borde de hueco; (c) Proyección del borde de hueco según la dirección de la normal al hueco.

Un ejemplo de bordes de hueco y su proyección sobre el plano de proyección, que se denotará por τ , se muestra en la figura.4.8(c) Con el fin de proyectar los vértices del borde de hueco sobre τ , se les aplica a ambos, vértices y plano, un cambio de sistema de referencia a Σ' , de tal manera que el plano XY' sea paralelo al plano τ .

- Orientación basada en la mejor aproximación de un planos a los vértices del borde de hueco

Un método para obtener el plano τ que mejor se adapta a una nube de puntos es aplicarle el análisis de las componentes principales (ACP). Este análisis construye una transformación lineal que escoge un nuevo sistema de coordenadas para la nube de puntos original en el cual la varianza de mayor tamaño de la nube es capturada en el primer eje (llamado el Primer Componente Principal o ACPEje1), la segunda varianza más grande es el segundo eje (llamado el Segundo Componente Principal o ACPEje2), y así sucesivamente. El plano que contiene a ambos ejes 1 y 2, es el que mejor se adapta a la nube de puntos. Este plano es otro de los posibles planos que pueden utilizarse como referencia para la obtención de la imagen de rango.

Para calcularlo, en primer lugar, se calculan las coordenadas de la nube de puntos con respecto a un sistema de coordenadas Σ' cuyos ejes X', Y' y Z' son paralelos a los ejes X, Y y Z iniciales. A continuación se realiza un análisis de las componentes principales en el nuevo sistema de referencia, de donde se obtienen el PCAEje1 y PCAEje2. La orientación del

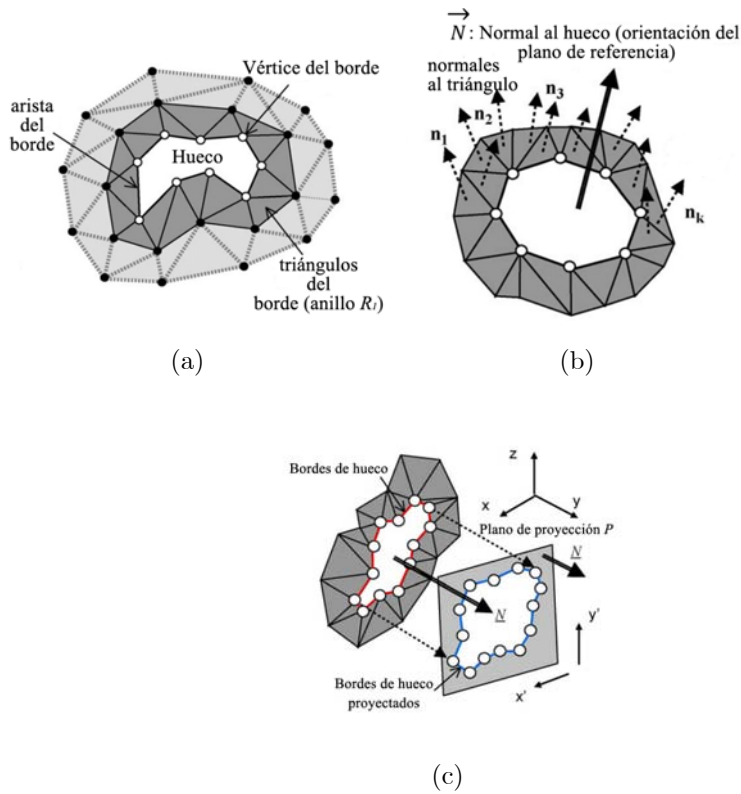


Figura 4.8: Three subfigures.

plano de referencia τ se obtendrá mediante el cálculo del vector normal a ambos ejes, que no es más que el producto vectorial de sendos vectores unitarios de cada eje. La dirección de la primera componente principal, $PC1$, pasa por el centroide $(0, 0, 0)$ y representa una línea de mínimos cuadrados que marca la tendencia dominante en los datos. La dirección del segundo componente principal, $PC2$, pasa por el centro de gravedad también, es ortogonal a $PC1$, y representa también una tendencia en los datos menos dominante, que es independiente de la tendencia capturada por el $PC1$.

- Orientación basada en el área máxima

El funcionamiento del método se basa en la aplicación de un algoritmo de restauración sobre la zona de la imagen en que se proyecta el hueco. Por ello, cabe pensar que será más conveniente fijar un plano sobre el que la proyección del hueco sea lo más representativa posible del hueco real sobre la malla, lo cual puede ser traducido en que la proyección ha de tener un área máxima. Por tanto, según este criterio de elección del plano de referencia, la mejor orientación es aquella en la que la proyección del borde del hueco B_k da lugar a un polígono P_k en el plano, cuya área sea máxima (figura 4.9).

Para resolver este problema se define un nuevo sistema de referencia $\Sigma'(X'Y'Z')$ ligado al plano de referencia τ , cuyos ejes X' e Y' estén contenidos en el mismo y cuyo eje Z' sea perpendicular a ambos, apuntando en dirección hacia el objeto. Seguidamente se plantea un problema de maximización, que buscará que Σ' o, lo que es lo mismo, τ tenga una orientación tal, que el área del polígono P_k sea máxima (figura 4.9).

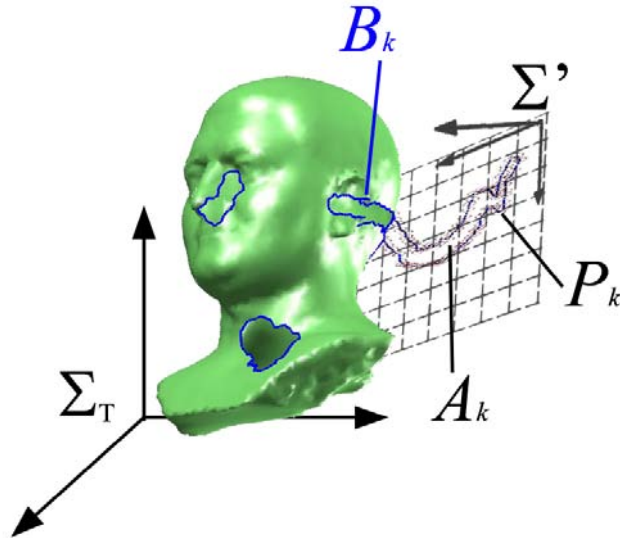


Figura 4.9: Cálculo de la orientación de ligado al plano de referencia, que haga que el Área de P_k , A_k , sea máxima

Por tanto, se plantea el problema de maximización siguiente:

$$\text{máx}[A_{P_k}(\Sigma')] \quad (4.11)$$

$$\text{sujeto a :} \quad (4.12)$$

$$|\hat{x}'| = 1$$

$$|\hat{y}'| = 1$$

$$(\hat{x}' \perp \hat{y}') \perp \hat{z}'$$

donde $A_{P_k}(\Sigma')$ es el área del polígono P_k contenido en el plano $X'Y'$ del sistema de referencia Σ' ; y \hat{i}' , \hat{j}' y \hat{k}' son los vectores unitarios de los ejes X' , Y' y Z' , respectivamente. Para la resolución del problema, se le estima un valor inicial de la solución igual a las componentes principales de los vértices v_{B_k} , calculadas según se explica en el criterio de orientación anterior. Con ello se facilita la consecución del valor máximo en un menor número de iteraciones. En la figura 4.10 puede observarse un ejemplo de aplicación del algoritmo de maximización de área en el borde del hueco 1 de la malla mostrada en la figura 4.2.

Posición del plano de proyección

Una vez obtenida la orientación del plano τ habría que aplicar una transformación T a la porción de malla M_{v_i} para obtener sus coordenadas desde el sistema de coordenadas $\Sigma'(X'Y'Z')$, solidario al plano. De esta forma, dadas las coordenadas de un punto de la malla (x_1, y_1, z_1) desde Σ_T , sus nuevas coordenadas (x_2, y_2, z_2) desde el sistema de referencia Σ' se obtendrían mediante la siguiente ecuación:

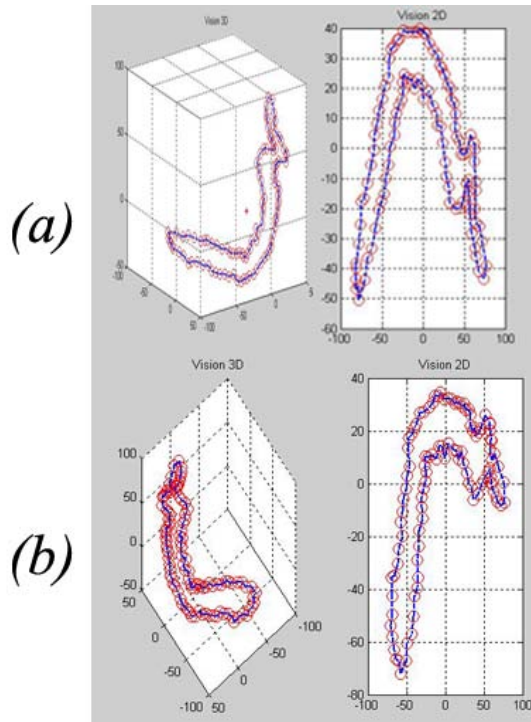


Figura 4.10: En (a) se muestra la representación 3D a la izquierda y la proyección sobre el plano de referencia inicial, del hueco 1. En (b) se muestra la representación 3D a la izquierda y la proyección sobre el plano de referencia que se ha obtenido tras aplicar el algoritmo de maximización del área.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{i}'_x & \hat{j}'_x & \hat{k}'_x & t_x \\ \hat{i}'_y & \hat{j}'_y & \hat{k}'_y & t_y \\ \hat{i}'_z & \hat{j}'_z & \hat{k}'_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \quad (4.13)$$

De esta transformación T queda aun por fijar la traslación que hay que aplicar a los puntos de la porción de malla M_{v_i} . Como se mencionó en el anterior capítulo, interesa que el plano de proyección no se encuentre a una distancia grande de la nube de puntos que va a proyectarse sobre él, ya que esto produciría que se almacenasen valores muy altos en cada píxel. La consecuencia inmediata que produciría esta situación es que el algoritmo de restauración necesite más iteraciones para alcanzar una solución óptima. Por tanto, ha de fijarse una posición del plano τ cercana a la malla M_{v_i} .

El criterio que se aplica para posicionar el plano toma como punto de referencia el centroide c_T de la malla total M_T , que se obtiene como sigue:

$$\begin{pmatrix} c_{Tx} & c_{Ty} & c_{Tz} \end{pmatrix} = \begin{pmatrix} \frac{\sum_{i=1}^K x_i}{K} & \frac{\sum_{i=1}^K y_i}{K} & \frac{\sum_{i=1}^K z_i}{K} \end{pmatrix} \quad (4.14)$$

siendo K el número de puntos que forman la nube $P_T \subset M_T$.

Seguidamente, dada la porción de malla M_{v_i} , se calculan las distancias del centroide c_T a todos los puntos de la nube $P_{v_i} \subset M_{v_i}$ como:

$$d(c_T, P_{v_i}) = \sqrt{(x_{v_i} - c_{Tx})^2 + (y_{v_i} - c_{Ty})^2 + (z_{v_i} - c_{Tz})^2} \quad (4.15)$$

A continuación se determinan los valores máximos y mínimos de esas distancias calculadas, $\max(d(c_T, P_{v_i}))$ y $\min(d(c_T, P_{v_i}))$, respectivamente. Se va a fijar la posición del plano τ a una distancia $\theta \max(d(c_T, P_{v_i}))$ o $\varphi \min(d(c_T, P_{v_i}))$ del centroide c_T , dependiendo si se desea situar ligeramente al exterior o al interior de la malla M_T , respectivamente. El valor de θ será mayor que la unidad y se elige normalmente en el intervalo (1, 1'5). En cambio φ será menor a la unidad y dentro del intervalo (0'5, 1). De esta manera, se establece que la posición del plano sea tal que contenga al punto $e(e_x, e_y, e_z)$. Este punto se trata del punto de intersección entre la recta que une los centroides c_T y c_v y la esfera de radio $\theta \max(d(c_T, P_{v_i}))$, centrada en c_T . La recta vendrá dada por la ecuación:

$$\frac{x - c_{Tx}}{c_{vx} - c_{Tx}} = \frac{y - c_{Ty}}{c_{vy} - c_{Ty}} = \frac{z - c_{Tz}}{c_{vz} - c_{Tz}} \quad (4.16)$$

y la esfera se define mediante la siguiente ecuación:

$$(x - c_{Tx})^2 + (y - c_{Ty})^2 + (z - c_{Tz})^2 = r^2 \quad (4.17)$$

en donde el radio r , como se dijo anteriormente, puede fijarse en $\theta \max(d(c_T, P_{v_i}))$ o $\varphi \min(d(c_T, P_{v_i}))$. Para acotar, se utiliza la siguiente ecuación en lugar de la anterior, para que se elija la solución de la intersección que esté más próxima al centroide de la porción de malla c_v :

$$\overrightarrow{c_T c_v} \cdot \overrightarrow{c_T e} = (c_{vx} - c_{Tx})(e_x - c_{Tx}) + (c_{vy} - c_{Ty})(e_y - c_{Ty}) + (c_{vz} - c_{Tz})(e_z - c_{Tz}) = r \cdot d(c_T, c_v) \quad (4.18)$$

la cual viene a decir que los vectores $\overrightarrow{c_T c_v}$ y $\overrightarrow{c_T e}$, además de ser paralelos debido a cómo se han definido, también tienen la misma dirección. Por tanto, tomando las ecuaciones 4.16 y 4.18, se obtienen las coordenadas del punto e como:

$$\begin{pmatrix} e_x & e_y & e_z \end{pmatrix} = \begin{pmatrix} \frac{r \cdot d(c_T, c_v) \cdot (c_{vx} - c_{Tx})}{\|\overrightarrow{c_T c_v}\|^2} + c_{Tx} & \frac{r \cdot d(c_T, c_v) \cdot (c_{vy} - c_{Ty})}{\|\overrightarrow{c_T c_v}\|^2} + c_{Ty} & \frac{r \cdot d(c_T, c_v) \cdot (c_{vz} - c_{Tz})}{\|\overrightarrow{c_T c_v}\|^2} + c_{Tz} \end{pmatrix} \quad (4.19)$$

Consiguientemente, la matriz de transformación que finalmente se utilizará tendrá como componente de traslación el vector $\begin{pmatrix} e_x & e_y & e_z \end{pmatrix}$, lo que da lugar a que las nuevas coordenadas se obtengan aplicando la siguiente ecuación:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{i}'_x & \hat{j}'_x & \hat{k}'_x & e_x \\ \hat{i}'_y & \hat{j}'_y & \hat{k}'_y & e_y \\ \hat{i}'_z & \hat{j}'_z & \hat{k}'_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \quad (4.20)$$

En la figura se muestra un ejemplo de determinación de la posición del plano de referencia τ .

Tamaño de la imagen de rango

Con respecto al tamaño de la imagen de rango, éste viene determinado por el tamaño de la porción de malla que se ha de proyectar.

A partir de esta nueva representación 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

Figura 4.11: Posición del plano de referencia con respecto a la porción de malla.

en X' y en Y' son x_{max} , y_{max} y x_{min} , y_{min} , respectivamente, el tamaño que representa la imagen es de $w \times h$, en donde:

$$w = x_{max} - x_{min} \quad (4.21)$$

$$h = y_{max} - y_{min} \quad (4.22)$$

Este tamaño se define de igual manera que como se definía para vistas parciales en la sección 3.4 del capítulo 3, ya que puede asimilarse la porción de malla obtenida a las vistas parciales que se empleaban en tal capítulo.

4.5.2. Resolución de la imagen de rango

Una vez que se ha determinado la orientación, posición y el tamaño del plano de proyección, el siguiente paso es la definición de la resolución de la imagen I_{h_i} en píxeles. La elección del ancho de celda de la rejilla sobre la que proyectar los puntos de M_{v_i} se hace de forma idéntica a cómo se hacía para vistas parciales. De esta forma, se ha de buscar una anchura de la rejilla óptima $q(mm/pixels)$, y con ella se tendrá que la resolución de la imagen es $1/q(pixels/mm)$ y su tamaño es $(n, m) = (w/q \times h/q)(pixels)$, como se deducía en el capítulo anterior.

En el caso de vistas parciales, ya se determinó la importancia que tiene en esta etapa que la malla presente una distribución más o menos uniforme. En cualquier caso, siempre puede estudiarse tal uniformidad antes de comenzar el proceso de relleno y, en situaciones de distribución no uniforme, aplicar un algoritmo de remallado.

Para determinar el ancho óptimo q , se aplican los estudios realizados con las vistas parciales. Se ha de recordar pues, que se definía el valor de q proporcional a la media de las longitudes de las aristas, d_m :

$$q = \alpha \cdot d_m \quad (4.23)$$

Para el cálculo de d_m podría plantearse en un primer lugar la conveniencia de realizarlo al comienzo del proceso para la malla global M_T , para evitar así la repetición del proceso de medidas de longitudes de arista en cada porción de malla. Sin embargo, esta opción puede dar lugar a que la imagen de rango se ajuste correctamente a las características de la malla M_{v_i} , dado que uno puede garantizarse una distribución perfectamente uniforme a lo largo de la malla global. Por tanto, para conseguir un mejor ajuste, se calcula la media de las longitudes de las aristas, d_{m_i} , para cada una de las porciones de malla, para otorgar más permisividad a los requisitos de uniformidad de la malla. Por consiguiente, se va a calcular una anchura de rejilla óptima, q_i , para cada hueco a rellenar, o también, un valor de resolución, $1/q_i$, para cada hueco h_i .

En cuanto al valor de α , dada la similitud entre las vistas parciales y las porciones de malla de la situación actual, por haberse escaneado ambas con el mismo equipo, se tendrá en cuenta toda la experimentación realizada con dichas vistas y se aceptarán los resultados empíricos y conclusiones que se obtuvieron en el capítulo 3.

Otro parámetro, que también fue tratado en el capítulo anterior, y que resulta igualmente aplicable en el actual es el factor de escala f_e . Anteriormente se explicó que, a la hora de fijar la posición del plano de referencia, se busca que la distancia entre y porción de malla sea pequeña, mediante la limitación de los valores de θ y φ dentro de determinados intervalos. Aun introduciendo estas limitaciones, también va a aplicarse un factor de escala, f_{e_h} , en los valores de distancias del plano a los puntos de M_{v_i} , para lograr, si cabe, un mayor acercamiento, más refinado, entre ambos y, por tanto, un menor número de iteraciones necesarias del algoritmo de relleno para conseguir un valor final adecuado. De esta forma, también se obtendrá la misma relación que se obtenía en el capítulo anterior:

$$I'_h = I_h \cdot f_{e_h} \quad (4.24)$$

Además, si bien los valores de θ y φ se fijaban de forma global para toda la malla M_T , la determinación de f_{e_h} dependerá de la máxima variación que se dé en los valores de la coordenada Z' en cada porción de malla M_{v_i} .

Siguiendo con el proceso, tras haber obtenido la imagen de rango, I'_h , se dispone de una representación de la porción del objeto 3D en formato de imagen al que se va a aplicar el algoritmo de restauración de imagen, para así obtener la imagen restaurada, I_f , que se está buscando. A continuación se van a describir los algoritmos de restauración de imagen que van a ser aplicados.

Interpolación de la imagen de rango

En el proceso de generación de la imagen de rango, también ocurrirá lo mismo que con las vistas parciales, que, al no contar generalmente con mallas con distribución totalmente uniforme, la imagen presentará las pérdidas en algunos píxeles. Estas pérdidas son debidas a un tamaño grande de algunos triángulos con respecto al ancho de celda, lo que se traduce en que puntos de esos triángulos se proyecten en puntos que no son vecinos en la imagen. Por tanto, aparecen en la imagen algunos píxeles vacíos lo que representa una pérdida de continuidad en la imagen en donde sí la hay en la superficie proyectada.

Además, hay que añadir una diferencia al trabajar con porciones de malla con respecto al caso más simple de las vistas parciales. En esta situación y tal y como ha sido explicado el proceso de formación de la imagen de rango, es posible que se añadan más pérdidas de información en la imagen de rango, debido a la orientación del plano de referencia. Esto es, puede ocurrir que debido a una elevada curvatura en algunas zonas de la malla (por ej. una

Figura 4.12: Aparición de pérdidas en la imagen de rango debidas a la orientación del plano de referencia con respecto a la malla.

concauidad o hendidura de cierta profundidad), sea imposible obtener una continuidad en la imagen de rango para una posición y orientación determinados del plano de referencia. En la figura 4.12 se reflejan el tipo de pérdidas que se ha mencionado.

Una vez detectadas en la imagen todas las pérdidas, se procederá a la interpolación de esas zonas de la misma forma que se explicó en el capítulo anterior, para restituir esa información temporalmente, y así tener más información disponible para el algoritmo de restauración.

Hay que señalar que si bien con vistas parciales, el proceso de relleno se aplicaba de golpe sobre todos los huecos que presentaban, o lo que es lo mismo sobre todos los huecos de la imagen de rango, con las porciones de malla no se va a proceder de la misma forma. Como ya se ha explicado, al inicio del proceso de relleno se buscan y detectan todos los huecos de M_T , y es para cada uno de los huecos para los que se extrae la porción de malla circundante. Por tanto, dada una porción de malla M_{v_i} , se tiene un único objetivo de rellenar el hueco h_i . Sin embargo, dada la complejidad que puede darse en la digitalización de objetos reales, es posible que en determinadas zonas aparezca más de un hueco en una misma zona de la malla. Por tanto, ante tal situación es probable que, si la distancia entre ambos es pequeña, esos huecos aparezcan en la misma porción de malla M_{v_i} , extraída para uno de ellos. Dado que en el proceso de generación de la imagen de rango solamente se tiene identificado sobre la imagen el hueco h_i correspondiente a esa porción de malla, el otro u otros huecos capturados en la imagen, van a ser representados como zonas vacías en la misma. Si, debido a la complejidad de la malla y orientación del plano de referencia, el hueco que no es el objetivo, h_{i+1} , es representado como una zona vacía pequeña, el algoritmo de interpolación lo detectará como una de las zonas de pérdidas y procederá a su relleno temporal. Si, por el contrario, el hueco se proyecta dando lugar a una zona vacía de tamaño considerable, esta no será interpolada y producirá una merma en la cantidad de información que tomará de la imagen el algoritmo de restauración. Esta situación puede producir un relleno de peor calidad del hueco h_i . A

Figura 4.13: Aparición de más de un hueco en la imagen de rango de uno de ellos

pesar de este inconveniente, dada la naturaleza iterativa del proceso de relleno de la malla global M_T , cuando en un futuro paso de iteración se acometa el relleno del otro hueco h_{i+1} , el escenario que se presente será algo más ventajoso. Si la correspondiente malla $M_{v_{i+1}}$ contiene la zona donde se localizaba h_i , al haberse relleno ya ese hueco, ahora no se tendrá el problema anterior. Aunque podría cuestionarse la validez de la información que aporte la información restaurada de h_i en el relleno de h_{i+1} , es más favorable esta situación para añadir más información, aunque aproximada, a la imagen que se le pasa al algoritmo de restauración. El caso que se ha explicado viene representado en la figura.

4.6. Aplicación de los algoritmos de image inpainting

En esta etapa se va a aplicar el algoritmo de image inpainting sobre la imagen $I_{\{h\}}$ obtenida en la etapa anterior. Como se mencionó anteriormente, una vez revisada la literatura sobre los algoritmos de restauración de imágenes, se decidió comprobar el funcionamiento del método incluyendo dos tipos de algoritmos de restauración, cada uno perteneciente a una de las categorías enumeradas. En concreto, se planteó la aplicación de un método basado en la difusión de información: el desarrollado por Criminisi et al. ; y otro método basado en la información estadística: el desarrollado por Roth y Black . En ambos se observó y comprobó la bondad de los resultados que producían al utilizarlos con fotografías, y por eso se decidió aplicarles en el método propuesto.

A continuación se describirá brevemente la base teórica de estos algoritmos para, seguidamente, explicar el modo de integración de esos algoritmos en el método que se pretende verificar en este capítulo.

Algoritmo de restauración de la imagen basado en el muestreo

Este método, descrito en , combina las técnicas de síntesis de textura que utilizan muestreo y las técnicas de difusión de imagen para calcular el relleno de huecos en la imagen.

Para este fin, el algoritmo comienza con la introducción por parte del usuario de la máscara de selección. Esta máscara indica qué zonas de la imagen han de ser restauradas. El usuario también tiene que especificar el ancho de la ventana que se utilizará para el muestreo. Aunque en general este parámetro suele fijarse en un valor que funciona bien en la mayoría de los casos.

Durante su ejecución, el algoritmo va rellenando, desde los bordes hacia el interior, los huecos seleccionados. Este relleno lo realiza estudiando la textura y la estructura de la imagen, tomando muestras y propagando la información de textura y estructura hacia el interior de la imagen.

Hay que resaltar que es muy importante el orden en que se realiza el relleno de los píxeles de la imagen. Esto es debido a que se trata de un algoritmo iterativo que va teniendo en cuenta el valor del píxel que se calculó en la iteración anterior. El método que emplean es el llamado piel de cebolla, que va rellenando anillos del hueco de forma concéntrica hacia el interior. En la puede observarse un ejemplo del funcionamiento de este algoritmo con una imagen en la que se ha querido eliminar uno de los elementos que aparecen.

[ejemplo_criminisi.tif]

[Ejemplos de reparación de imágenes con el algoritmo propuesto por Criminisi et al. en

Algoritmo de restauración basado en el Producto de Expertos

Esta técnica, desarrollada por , explota las ideas de la codificación de imágenes dispersa (sparse image coding) que consigue representar parches de la imagen como la combinación lineal de filtros aprendidos para el aprendizaje del comportamiento estadístico de las imágenes, utilizando los Campos de Markov (MRF- Markov Random Fields).

El método empleado se denomina Campo de Expertos y modela la probabilidad de una imagen en términos de un campo aleatorio con cliques (parches con determinadas relaciones de vecindad) solapados, cuyos potenciales son representados como un Producto de Expertos

La función del Producto de Expertos es modelar distribuciones de probabilidad de elevadas dimensiones (parches de la imagen) a partir del producto de varias distribuciones de expertos, donde cada experto trabaja en un subespacio de baja dimensión, sencillo de modelar. En concreto, al aplicar un filtro lineal a un parche de imagen se produce el subespacio de una dimensión que se modelará. Por tanto, la densidad de probabilidad del parche de imagen puede ser definido como el producto de las distribuciones que describen los filtros lineales.

Para modelar la probabilidad de la imagen completa, se emplea el Campo de Markov que modela la probabilidad de la imagen como una distribución de Gibbs $p = (1/z) e^{-\sum_k V_k(x_k)}$, donde x es una imagen y $V_k(x_k)$ es la función potencial para el clique x_k . Considerando un Campo de Markov homogéneo, todos los potenciales V_k son los mismos para todos los cliques, $V_k(x_k) = V(x_k)$. Para definir la función V , los potenciales del Campo de Markov son representados como el Producto de Expertos para todos los cliques. Consecuentemente, la probabilidad de la imagen puede ser modelada en términos del producto de las distribuciones de filtros lineales.

Todos los parámetros en este modelo son aprendidos a partir de un conjunto de imágenes, maximizando su probabilidad, es decir, maximizando la probabilidad para los modelos del Producto de Experto y del Campo de Expertos.

El algoritmo propaga la información utilizando el Campo de Expertos dentro del conjunto de puntos pertenecientes a la vecindad de los píxeles que se desean rellenar. La propagación se realiza de forma iterativa, teniendo en cuenta el modelo de probabilidad de la imagen

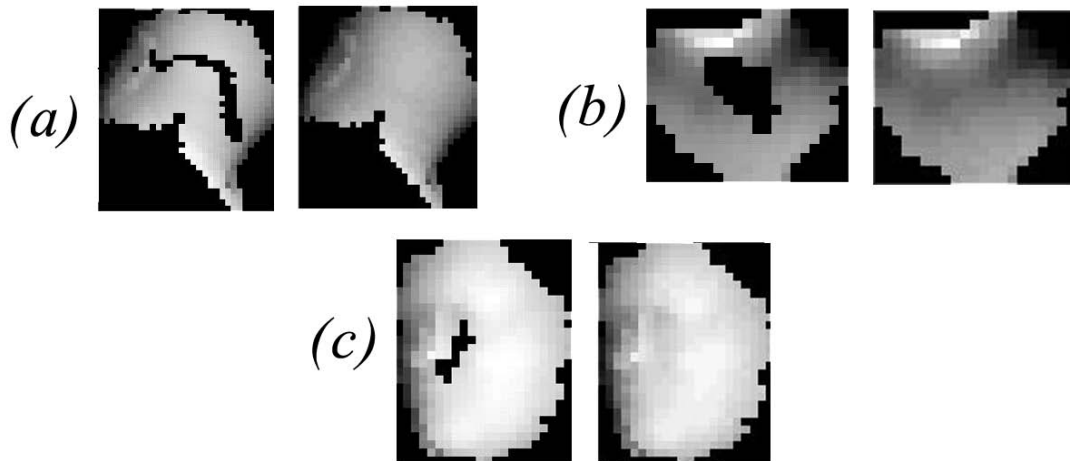


Figura 4.14: Representación de las imágenes de rango extraídas de las porciones de malla M_{v_i} (a la izquierda) y de las imágenes de rango obtenidas tras la aplicación del algoritmo de restauración, para los huecos 1, 2 y 3 en a , b y c , respectivamente.

(modelo de Gibbs). Como puede observarse en la llama la atención la eficacia del algoritmo en la restauración de las imágenes.

Este modelo ha sido entrenado utilizando 2000 regiones de imágenes que se han extraído de la base de datos para segmentación de Berkeley Segmentación (escenas naturales, personas, edificios, etc). En la Tabla se muestra un grupo de imágenes pertenecientes a esa base de datos.

[ejemplo_roth.tif]

[Ejemplos de reparación de imágenes con este algoritmo: (a) y (c) imágenes originales (dañada y con texto, respectivamente); (b) y (d) imágenes reparadas]

[plot]

Aplicación del algoritmo de restauración a la imagen de rango

En general este tipo de algoritmos no hace una selección automática de los píxeles sobre las que se efectuará el relleno. Solamente actuarán sobre los que les vengam marcados a la entrada. Por tanto, para la aplicación de estos algoritmos habrá que introducir tanto la imagen de rango, I_h , como una máscara de selección de los píxeles a rellenar, Q_h . Esta máscara es una imagen binaria, del mismo tamaño que la imagen de rango, $(n, m) = (w/q \times h/q)(pixels)$, y en las que aparecen activos los píxeles sobre los que se pretenda realizar el relleno, como se muestra en la figura ejemplo de mascara de seleccion.

[ejemplo de mascara de seleccion]

[Ejemplo de Máscara de selección]

4.7. Integración de los resultados. Transformación de 2D a 3D. Triangulación de huecos

De igual forma que se procedía en el capítulo anterior, la última etapa del proceso de relleno es la transformación de los datos que se han obtenido en la etapa anterior a 3D y su

integración en la malla global M_T . Evidentemente la transformación se aplica solamente a los datos nuevos que se han generado tras la aplicación del algoritmo de restauración, es decir, sobre los huecos rellenos, sin incluir en este grupo a los píxeles que han sido rellenos por el algoritmo de interpolación de la imagen de rango. Para seleccionar estos píxeles rellenos se aplica la máscara de selección a la imagen que resulta tras la restauración, como ya fue explicado en el relleno de vistas parciales.

De la misma manera, se realiza de forma idéntica la obtención de la coordenada 3D que se corresponde con esos píxeles rellenos. En este caso, teniendo en cuenta la ecuación 4.20, y el tamaño de la imagen definido en la ecuación ??, dado un píxel (r, s) que almacena el valor λ , la coordenada 3D, (x_f, y_f, z_f) , que correspondería a ese píxel es:

$$\begin{pmatrix} x_f \\ y_f \\ z_f \\ 1 \end{pmatrix} = \begin{pmatrix} \widehat{i}'_x & \widehat{j}'_x & \widehat{k}'_x & e_x \\ \widehat{i}'_y & \widehat{j}'_y & \widehat{k}'_y & e_y \\ \widehat{i}'_z & \widehat{j}'_z & \widehat{k}'_z & e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x_{\text{mín}} + s \cdot q \\ y_{\text{máx}} - r \cdot q \\ \frac{\lambda}{f_{e_k}} \\ 1 \end{pmatrix}; \quad r = \{1, 2, \dots, n\}; \quad s = \{1, 2, \dots, m\}$$

(4.25)

donde f_{e_k} es el factor de escala que se aplicó en la ecuación 4.24 para acercar los valores de la imagen de rango al nivel de cero en Z' (al plano $X'Y'$). Como ya se mencionó, este acercamiento ya se realizaba en mayor medida cuando se calculaba el vector de traslación (e_x, e_y, e_z) , por lo que, por término general, el factor de escala será de menos intensidad que el que se aplicaba en el capítulo anterior, es decir, muy próximo a la unidad. Además, en este caso este factor va depender más del proceso de selección del plano de referencia, siendo, por tanto específico para cada iteración del proceso de relleno (o para cada porción de malla seleccionada). En el capítulo anterior solo dependía del proceso de adquisición de la vista parcial, ya que las coordenadas del plano de referencia venían fijadas por la elección que se hiciese de la colocación del escáner 3D.

Hay que señalar, de nuevo, que en la ecuación 4.25, se considera que las coordenadas de los píxeles vienen definidas por la fila r y columna s que ocupan en la matriz imagen. Es por esto que que la coordenada x depende de la segunda coordenada del plano imagen, s , y la coordenada y de la primera, r .

Tras haber obtenido todas las coordenadas de los píxeles rellenos, se dispone de una nube de puntos 3D, R_i , correspondiente al hueco h_i de la malla que ha sido relleno. A continuación, es necesario definir las superficies rellenas mediante la triangulación de esas nubes de puntos. Este proceso se llevará a cabo de la misma forma que la explicada para vistas parciales. La única diferencia que hay que recordar es que en el proceso de relleno explicado en el capítulo anterior, éste se aplicaba simultáneamente sobre todos los huecos de la vista parcial. Por consiguiente, la nube de puntos que se obtenía tras la transformación de 2D a 3D, contenía puntos de los diferentes huecos rellenos, por lo que había que dividir la triangulación en grupos de puntos correspondientes a cada hueco, que estaba delimitado por su correspondiente borde de hueco. En el caso estudiado en este capítulo, sólo se va a rellenar un único hueco en cada triangulación, con lo que supone una simplificación del problema detallado en el anterior capítulo.

Para calcular la triangulación también va a incluirse en la nube de puntos, los puntos que forman el borde del hueco B_{h_i} , que servirá como delimitadores de la porción de malla rellena. Además, también tendrán la función de servir como eje de unión o cosido entre la malla inicial con huecos, M_T , y la malla de relleno.

En cuanto a la integración de la malla rellena, la tupla que caracteriza a la porción de

mallla rellena correspondiente al hueco h_i , $\{\mathcal{V}_i, \mathcal{A}_i, \mathcal{T}_i\}$, se adjuntará a continuación de la tupla de la malla global M_T , con lo que se obtendrá la tupla:

$$\left\{ \left[\begin{array}{c} \mathcal{V}_T \\ \mathcal{V}_i \end{array} \right], \left[\begin{array}{c} \mathcal{A}_T \\ \mathcal{A}_i \end{array} \right], \left[\begin{array}{c} \mathcal{T}_T \\ \mathcal{T}_i \end{array} \right] \right\} \quad (4.26)$$

La nueva malla global M'_T será la que se utilice en la siguiente iteración del proceso para proceder al relleno del hueco siguiente h_{i+1} , por lo que la malla global del proceso es diferente en cada iteración, pues va siendo poco a poco restaurada. En cualquier caso, resulta de especial interés mantener siempre identificadas cuáles eran las partes de la malla originales, y cuáles han sido generadas artificialmente en el proceso de restauración. Por ello, conviene asociar al proceso total la llamada matriz de restauraciones M_ρ , que tiene tres columnas y tantas filas como huecos se procesen. En las tres columnas se almacenan los cardinales de los conjuntos de vértices, aristas y triángulos que van añadiéndose en cada iteración del proceso, respectivamente. Con ello se tendrá la siguiente matriz de restauraciones:

$$M_\rho = \begin{bmatrix} \text{card}(\mathcal{V}_T) & \text{card}(\mathcal{A}_T) & \text{card}(\mathcal{T}_T) \\ \text{card}(\mathcal{V}_1) & \text{card}(\mathcal{A}_1) & \text{card}(\mathcal{T}_1) \\ \text{card}(\mathcal{V}_2) & \text{card}(\mathcal{A}_2) & \text{card}(\mathcal{T}_2) \\ \vdots & \vdots & \vdots \\ \text{card}(\mathcal{V}_m) & \text{card}(\mathcal{A}_m) & \text{card}(\mathcal{T}_m) \end{bmatrix} \quad (4.27)$$

En la figura 4.15 se muestra un ejemplo de triangulación e integración de tres huecos (a), (b) y (c), de la malla que se mostraba en la figura 4.2.

4.8. Resultados experimentales

4.9. Conclusiones

En este trabajo se ha presentado un método para el relleno de huecos en mallas 3D. Para esto es necesario, en primer lugar, generar una imagen 2D como entrada del algoritmo de restauración. Debido a la influencia de la resolución de la rejilla (pixels) en la pérdida de datos cuando los datos 3D son proyectados, se ha determinado la resolución óptima para minimizar esa pérdida.

Se han establecido dos comparaciones para estudiar las diferencias entre los diferentes métodos: la distancia entre las mallas originales y reparadas (por malla completa y por zona reparada) y la efectividad con respecto al área del hueco. Dos conclusiones importantes se pueden obtener. La primera es que el método propuesto ofrece errores más pequeños al utilizar el algoritmo de Roth, que en los otros casos. La segunda es que se ha probado que este es también más adecuada para rellenar huecos de cualquier tamaño. Por tanto, se ha demostrado que se trata de un método de relleno de huecos robusto.

En el futuro se pretende desarrollar trabajos centrados en el uso de información geométrica y de textura, como datos de apoyo para el algoritmo de relleno de huecos.

Por otra parte, el método ha sido probado solamente con vistas parciales aunque se está trabajando en la variación de la proyección de M_h de forma que se pueda aplicar a mallas completas.

Lineas de trabajo futuras

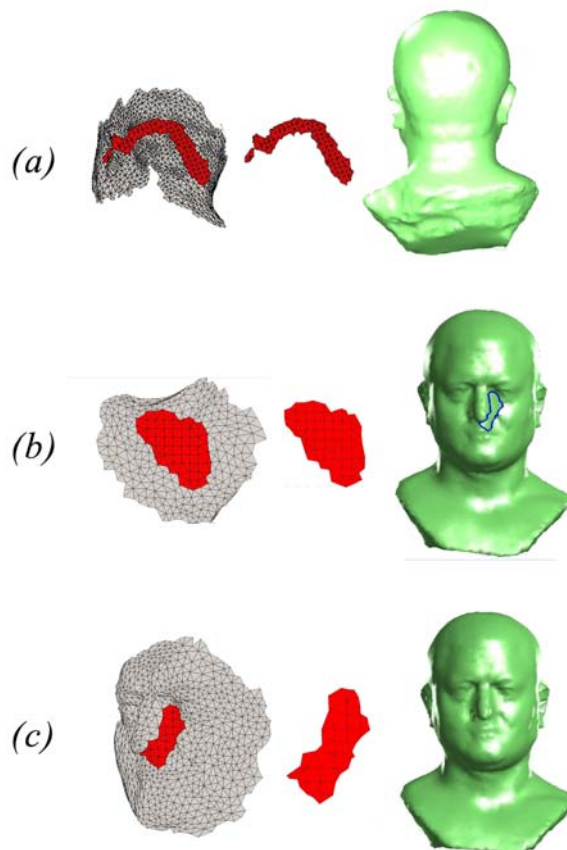


Figura 4.15: Integración de las zonas que se han rellenado aplicando el algoritmo de restauración en la malla total M_T de la figura 4.2. Huecos 1, 2 y 3 en (a), (b) y (c), respectivamente.

En este campo se han hecho pruebas de un nuevo parámetro en el método del cálculo del campo de distancias para reducir si cabe aún más los tiempos de computo. La idea es de definir un parámetro ■ que mide las diferencias que hay en la distancia calculada al moverse de un nivel a otro del árbol. De tal forma que, si la diferencia al bajar de nivel, se hace muy pequeña, se pararía el recorrido del árbol en ese nivel y se fijaría como distancia la última calculada.

Por otra parte, se pretende probar los algoritmos de restauración de imágenes sobre las secciones que se obtienen del campo distancias. Con esto pretende demostrar la validez del método de rellanado de huecos para este tipo de representación.

Por último, se fija como objetivo la creación de modelos para usarlos en el reconocimiento y localización 3D, a partir de las mallas restauradas, haciendo uso mixto de la geometría y textura. Para ello se combinarán ideas de los modelos existentes que hacen uso bien de la geometría del objeto o bien de su textura.

Capítulo 5

Conclusiones

En esta tesis se ha presentado un método para el relleno de huecos en mallas 3D. La idea novedosa que se proponía era la utilización de técnicas de restauración de fotografías, aplicadas sobre datos 2D en la resolución de un problema que se presentaba en objetos 3D. En primer lugar se ha tratado de validar el método propuesto mediante su aplicación sobre una base de datos de vistas parciales.

Una de las claves del método consiste en una transformación de los datos 3D en 2D para así poder aplicar esas técnicas de restauración de imágenes. Por tanto se ha procedido a generar una representación 2D de la malla que se quería rellenar, mediante la proyectación de los puntos 3D sobre un plano de referencia, lo que genera la correspondiente imagen de rango. Se ha comprobado la influencia que tiene en el proceso la etapa de generación de la imagen de rango y, en concreto, la elección de un ancho óptimo de la rejilla que constituye el plano de proyección y que determina el contenido de los píxeles de la imagen de rango que se genera. Por ello se ha hecho necesario la realización de un análisis experimental para determinar cuál es el ancho óptimo que se ha de fijar.

De entre diversas técnicas de restauración de fotografías se han elegido dos que se han considerado que ofrecían unos resultados muy buenos en su aplicación sobre diversos tipos de restauración. A partir de ahí se han generado resultados, de vistas parciales con huecos, rellenados aplicando sendas técnicas. Además para tener otra fuente de comparación se ha utilizado un software comercial para el relleno de huecos.

Para el análisis de todos los resultados se han establecido dos comparaciones para estudiar las diferencias entre los diferentes métodos: la distancia entre las mallas originales y reparadas (por malla completa y por zona reparada) y la efectividad con respecto al área del hueco. A partir de las mismas se han extraído dos importantes conclusiones. La primera es que el método propuesto ofrece errores más pequeños al utilizar el algoritmo de Roth, que en los otros casos. La segunda es que se ha probado que este es también más adecuada para rellenar huecos de cualquier tamaño. Por tanto, se ha demostrado que se trata de un método de relleno de huecos robusto.

Para su aplicación en mallas completas, se han estudiado las particularidades que se presentaban en tales condiciones.

En primer lugar las condiciones de la malla de partida son diferentes. Mientras que las vistas parciales no son apenas tratadas tras su digitalización, la malla de los objetos completos son el resultado de la aplicación de las diferentes etapas que componen el proceso de reconstrucción de objetos completos. Por tanto, la regularidad de las mallas en el primer caso es mucho mayor en el segundo, con los consiguientes procesos de remallado que ello requiere.

Además, por la misma razón, se hace necesario un análisis de los diferentes tipos de errores que pueden presentar las mallas de objetos completos. Se han establecido los mecanismos de corrección de dichos errores con el fin de garantizar un determinado nivel de calidad en la representación de la malla triangular y así poder aplicar los métodos de relleno sin apenas fuentes de error.

Una de las principales diferencias a la hora de trabajar con vistas parciales y objetos completos, es en la etapa de elección de plano de referencia que se utilizará para generar la imagen de rango. Si bien en el caso de vistas parciales, el plano de referencia no requiere de ningún cálculo previo ya que viene determinado por la posición del escaner cuando se digitalizó el objeto, en objetos completos esto genera un nuevo problema a resolver. Para ello, se han estudiado diversos métodos de elección del plano de referencia idóneo. En concreto se ha optado por un método que se basa en el cálculo de la orientación de tal plano que dé lugar a una proyección de área máxima del polígono genera el borde de hueco (contorno cerrado).

Esto pone de relevancia otra diferencia importante, si en vistas parciales se utilizaba un único plano para rellenar todos los huecos, en objetos completos habrá que calcular el plano óptimo para cada uno de los huecos que se presenten.

De esto último también puede extraerse otra diferencia más, en vistas parciales se genera una única imagen de rango, el relleno de los diversos huecos durante la etapa de restauración de imagen se hace de una vez, y por tanto, una vez trianguladas las correspondientes nubes de puntos de relleno, la integración se hace también a la vez, para todos los huecos rellenos. Sin embargo, en objetos completos, se genera una única imagen de rango para cada uno de los huecos y, por tanto, el proceso de restauración de la imagen de rango da lugar al relleno de un único hueco y, aun más, tras la triangulación de la nube de puntos, la integración de la malla rellena con la malla total se realiza por separado para cada uno de los huecos.

Tras el análisis del problema sobre objetos completos, también se ha diseñado un método de interpolación de la imagen de rango, para evitar el uso de imágenes de rango con un gran porcentaje de errores en su generación.

Dada la diversidad de tipos de huecos que pueden presentarse, con gran variación en el tamaño, curvatura y complejidad de los mismos, se ha efectuado un análisis para acotar los límites en que el comportamiento del método desarrollado ofrece buenos resultados y de esta manera determinar cuáles son las limitaciones que se han encontrado en su aplicación.

Para el análisis de los resultados se han calculado las curvaturas de las mallas rellenas y se ha observado la continuidad que presenta la superficie en las zonas rellenas y zonas de unión entre la malla de partida y las zonas restauradas. Además para cada experimentación se ha calculado el área del hueco relleno y los tiempos de relleno, con el fin de tener una base que sirva de referencia para su comparación con otros métodos de relleno existentes.

5.1. Líneas de trabajo futuras

Una vez concluidos los trabajos que se propusieron una vez que surgió la idea de esta tesis, se estudió la posibilidad de extender su aplicación en otros casos particulares del relleno en 3D. En concreto, se comenzó, y se continua actualmente, con la adaptación del/de los algoritmo/s para trabajar con datos generados por sensores de larga distancia. El problema surge cuando se planteó un proyecto para la obtención de mapas actualizados reales de interiores de edificios, tras la digitalización de los mismo. En ese proceso se obtenían lógicamente huecos en las paredes, suelos y techos digitalizados. En este caso el origen de los huecos es debido a causas diversas tales como: oclusión de los elementos (paredes, techos y suelos)

debido al mobiliario, falta de datos debido a una mala estrategia de escaneado, brillos o superficies oscuras, etc. Para esta situación particular, se plantea el estudio de los siguientes aspectos:

- Filtrado y adaptación del método para este tipo de datos.
- Estudio de los resultados según el tamaño de los huecos.
- Adaptación del algoritmo para el rellenado de casos particulares tales como: cenefas, marcos, etc.

Por otra parte, se ha estado investigando el algoritmo de restauración de imágenes empleado. Como ya se explicó, el funcionamiento se basa en el aprendizaje de patrones de respuesta a filtros de diversas porciones de imagen extraídas al azar de un banco de imágenes naturales. Se ha estado desarrollando una aplicación para el entrenamiento de este algoritmo utilizando cualquier fuente de imágenes. La idea en la que se pretende trabajar en un futuro próximo es en la realización de un banco de imágenes de rango obtenidas de diversos objetos reales y a partir de ahí, llevar a cabo el entrenamiento del algoritmo con ese nuevo banco de imágenes. A continuación se realizará un estudio de los resultados que ofrece el método propuesto en esta tesis, pero con ese nuevo entrenamiento aplicado.

Capítulo 6

Anexo I

Tabla 6.1: Generación de huecos artificiales en 20 vistas parciales de la base de datos

N Malla	Generación de huecos artificiales			Huecos numerados
	Malla original	Malla con huecos generados		
1				
2				

Tabla 6.1: (continuación)

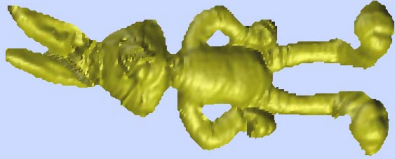
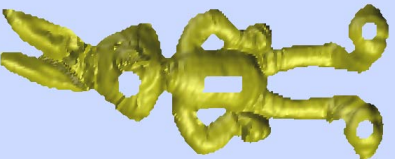
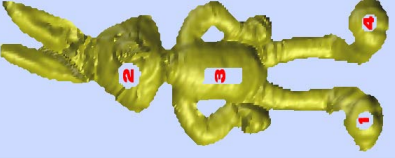
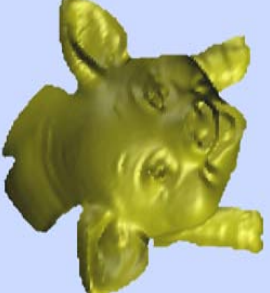
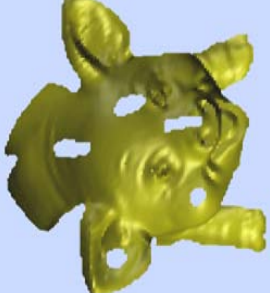
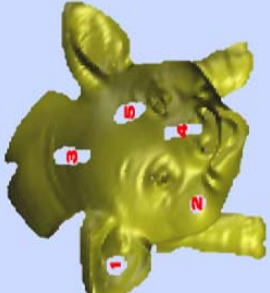
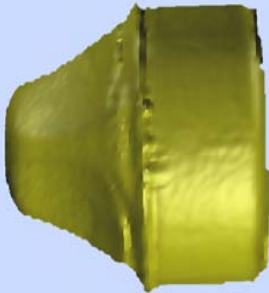
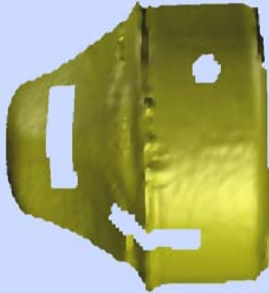
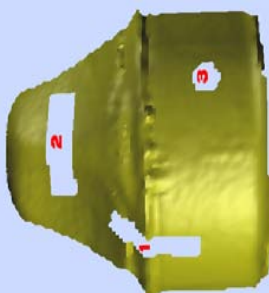
3			
4			
5			

Tabla 6.1: (continuación)

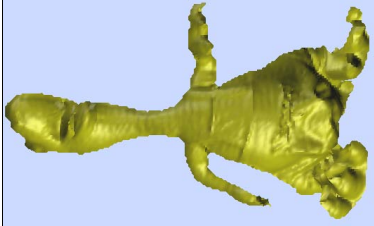


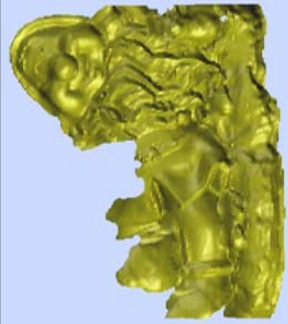
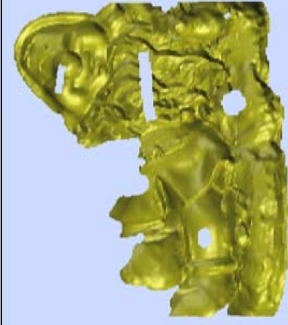
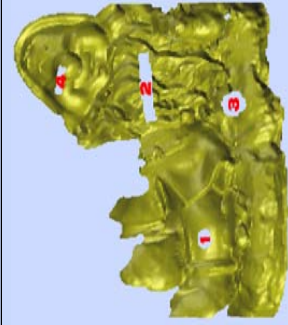
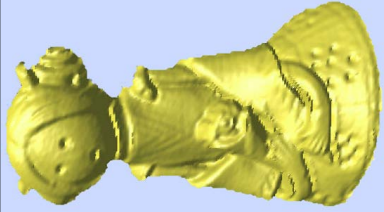
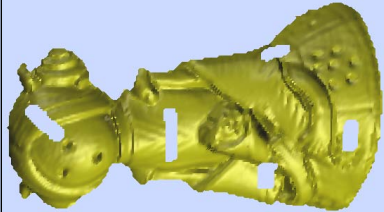
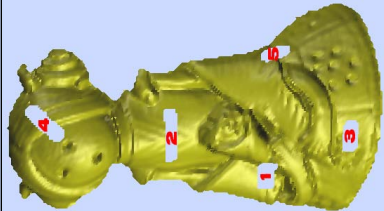
6			
7			
8			

Tabla 6.1: (continuación)



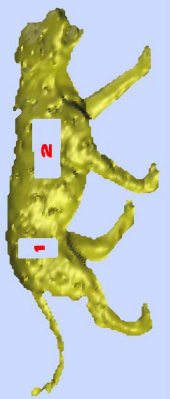
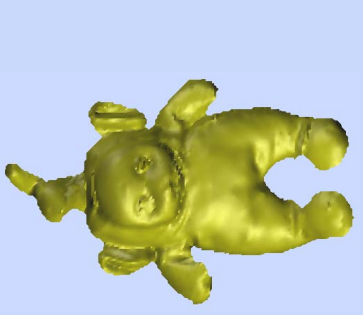
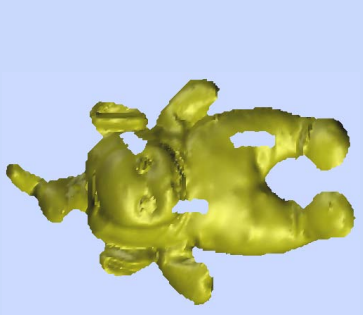
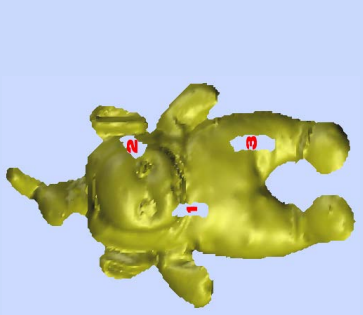
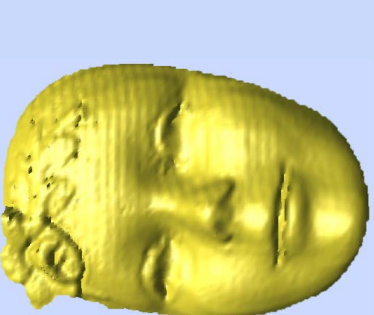
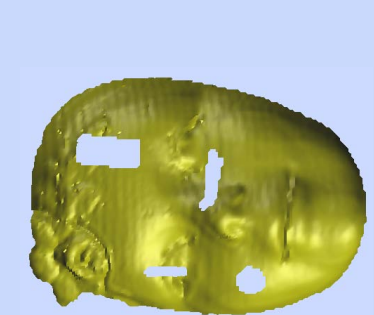
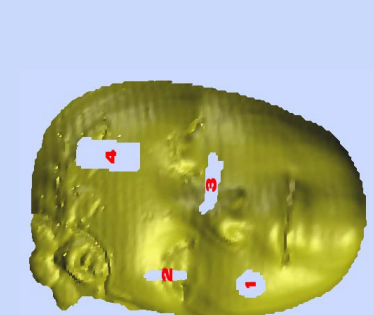
9			
10			
11			

Tabla 6.1: (continuación)


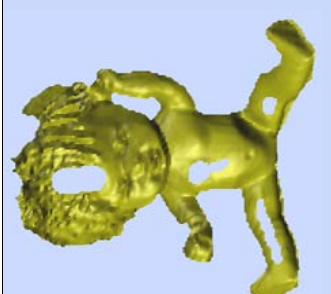
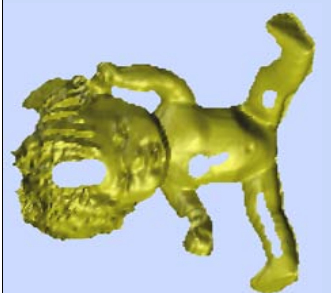
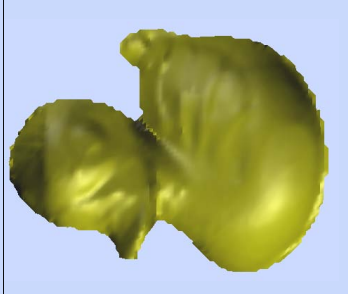
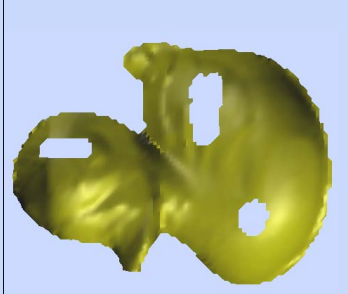
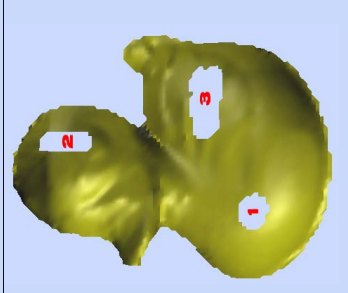
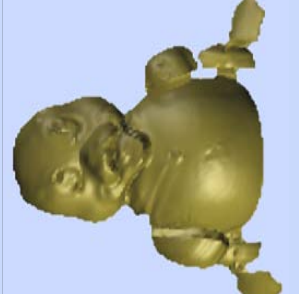
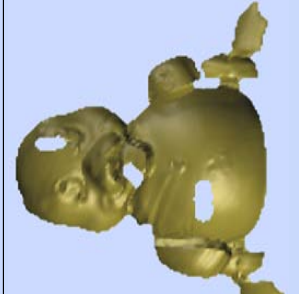
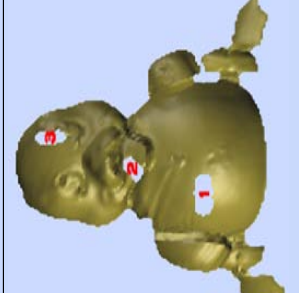
12			
13			
14			

Tabla 6.1: (continuación)

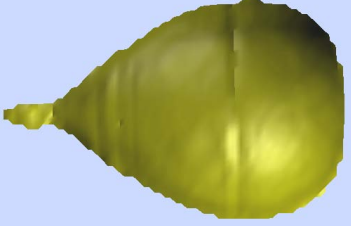

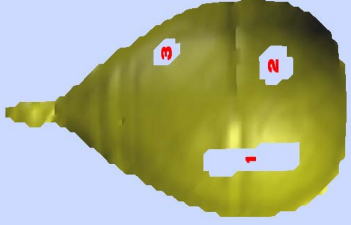
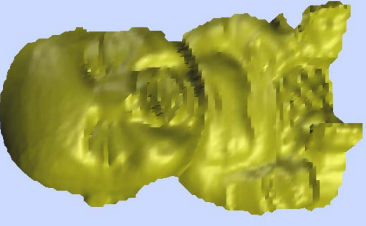
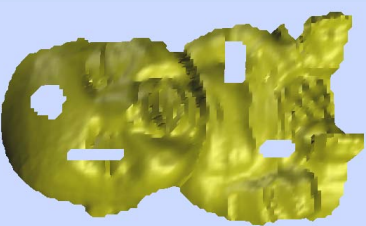
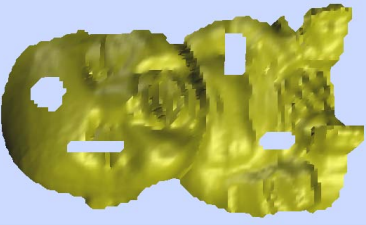
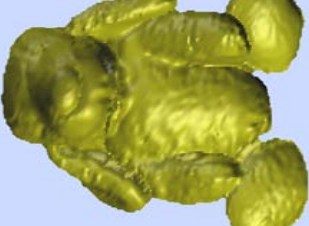
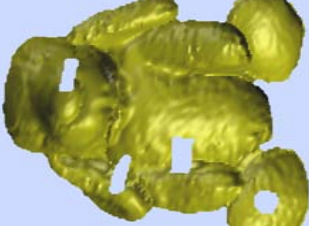
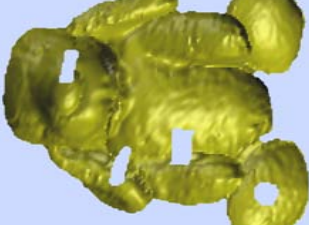
15			
16			
17			

Tabla 6.1: (continuación)

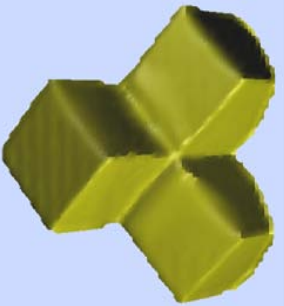
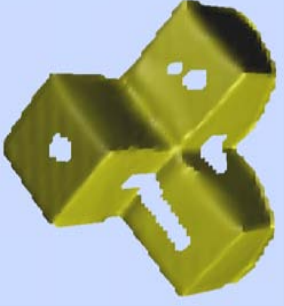
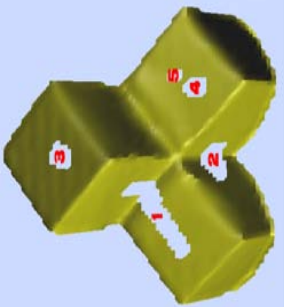
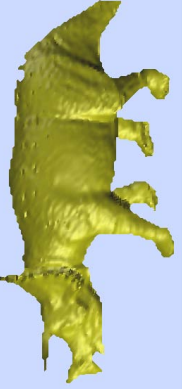
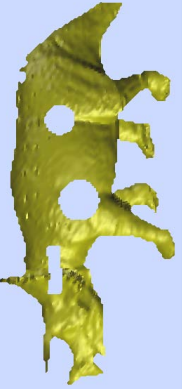
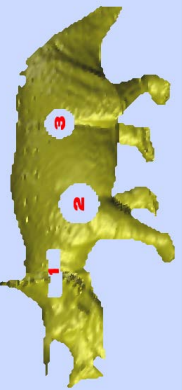
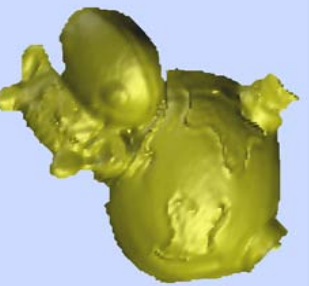
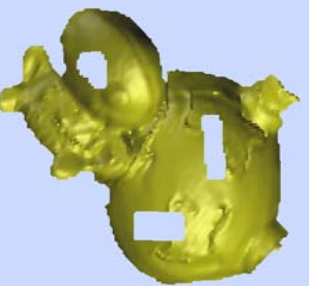
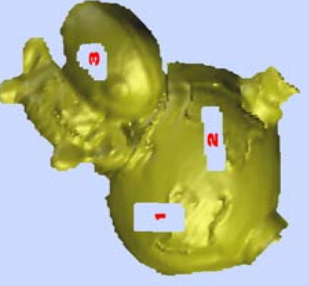
18			
19			
20			

Tabla 6.2: Comparación de algunas mallas rellenas con cada uno de los tres algoritmos aplicados

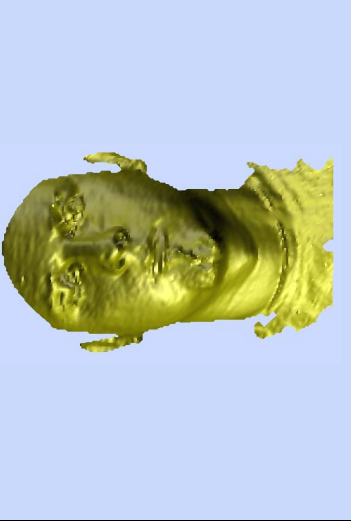
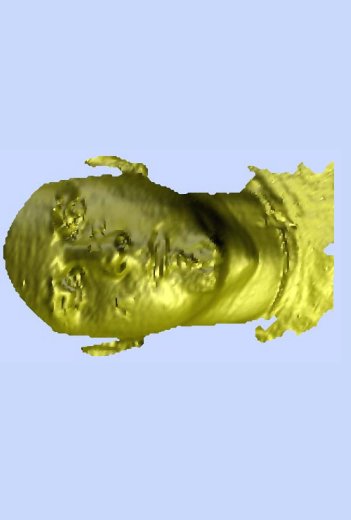
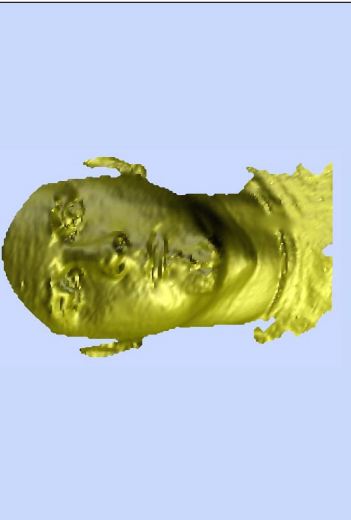
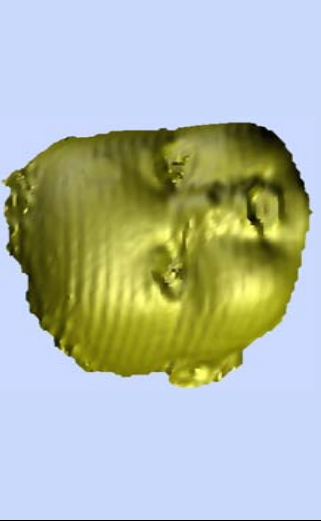
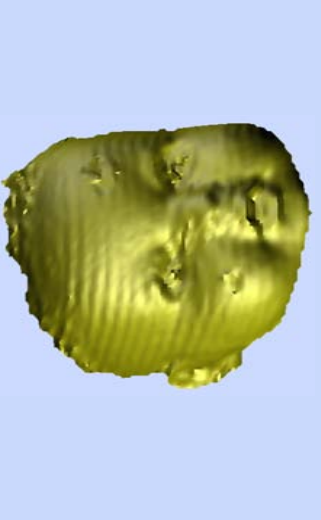
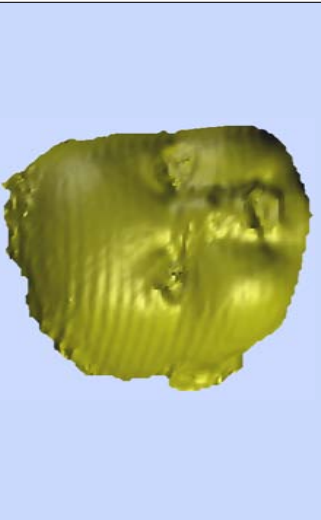
Comparación de mallas rellenas por algoritmo			
N Malla	Algoritmo 1	Algoritmo 2	Algoritmo 3
1			
2			

Tabla 6.2: (continuación)

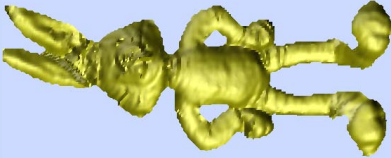
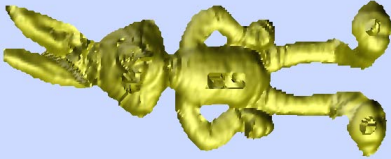
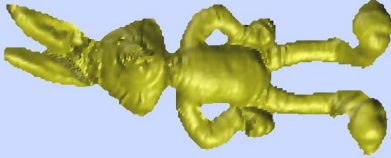

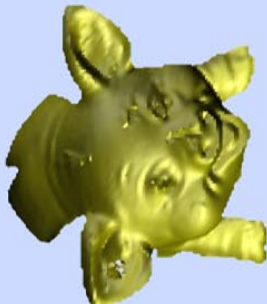

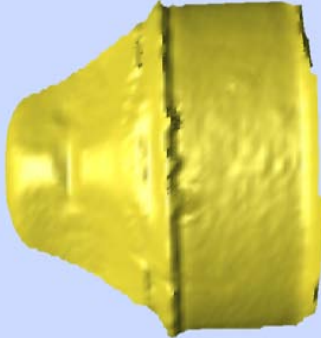
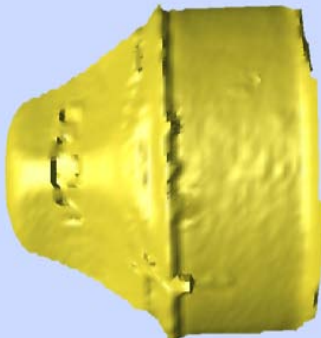
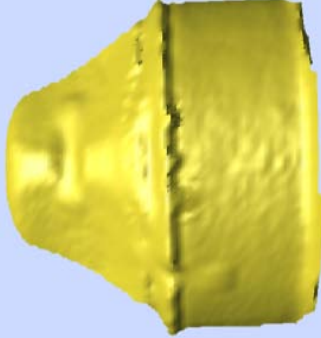
3			
4			
5			

Tabla 6.2: (continuación)




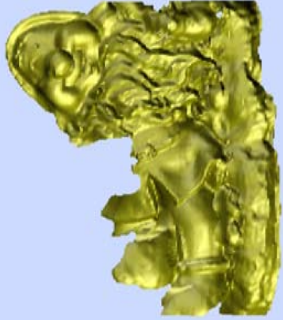
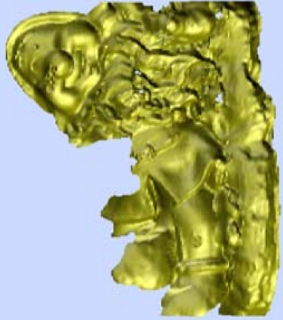
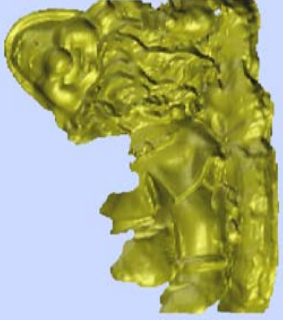
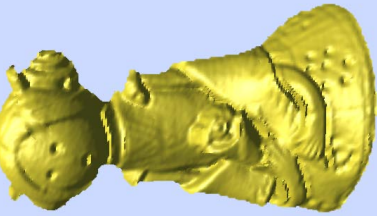
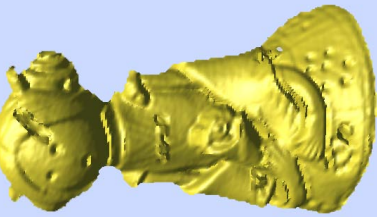
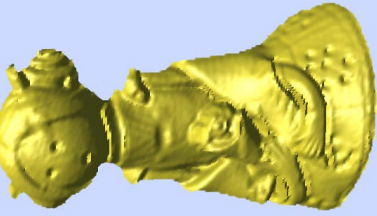
6			
7			
8			

Tabla 6.2: (continuación)

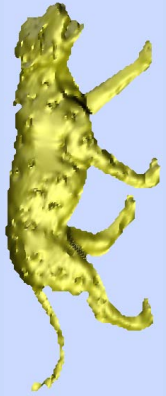
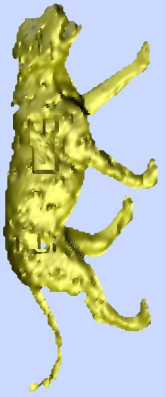
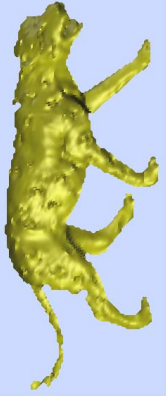
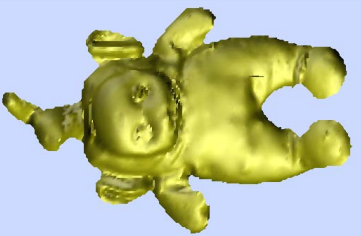
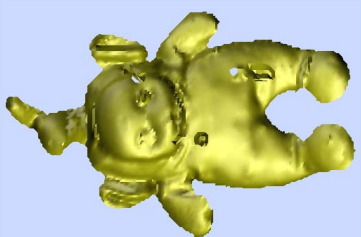
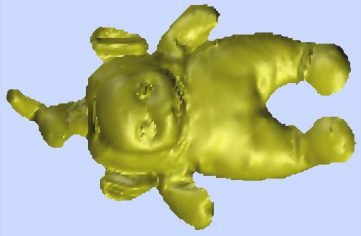
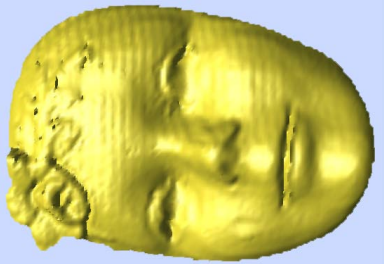
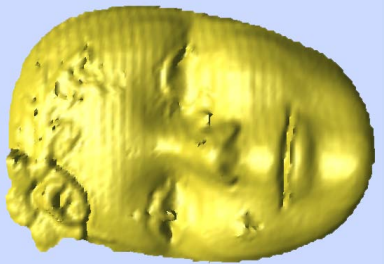
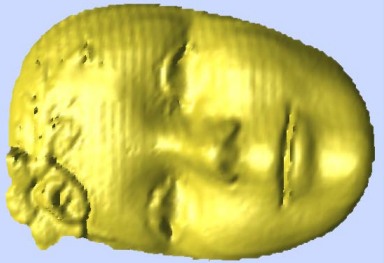
9			
10			
11			

Tabla 6.2: (continuación)


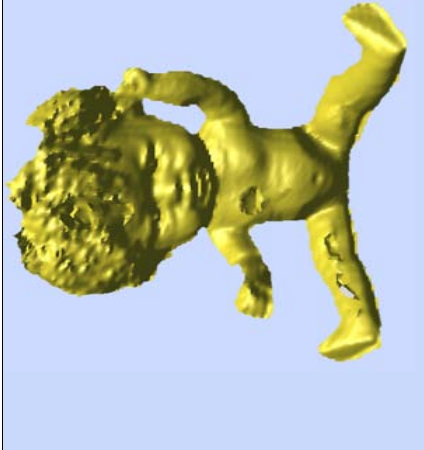
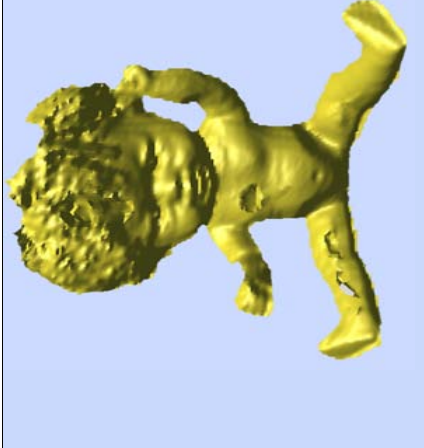
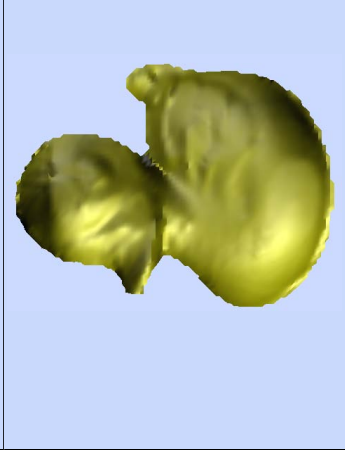
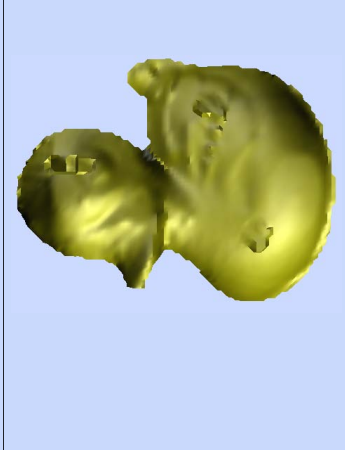
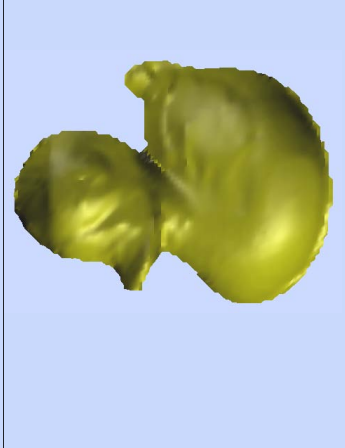
12			
13			

Tabla 6.2: (continuación)



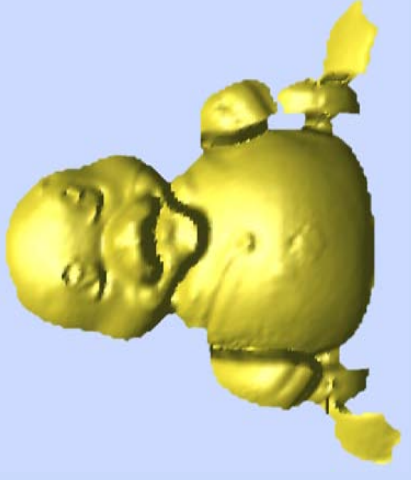
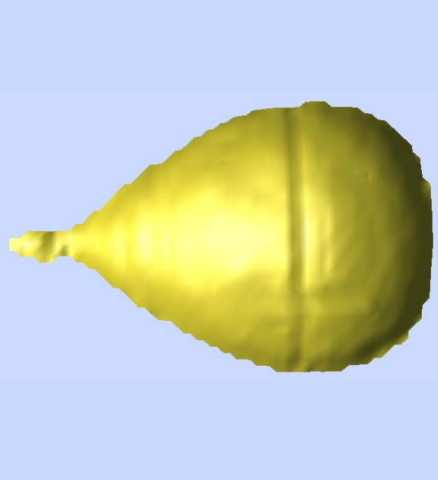
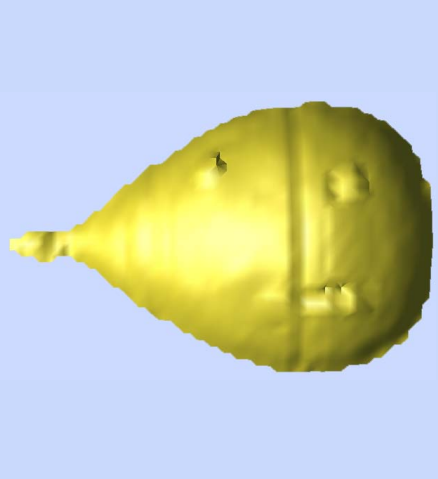
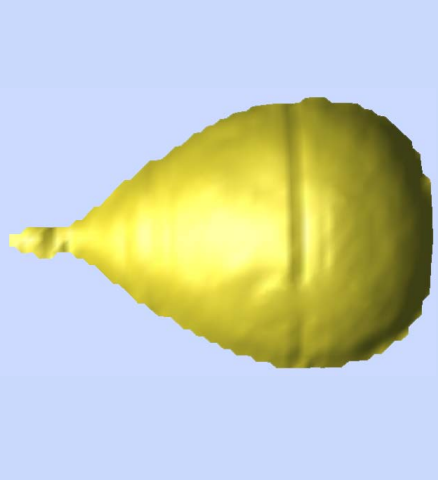
14			
15			

Tabla 6.2: (continuación)

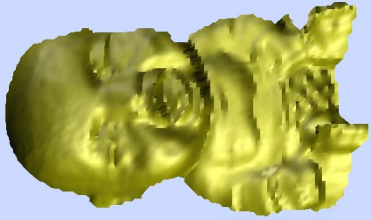
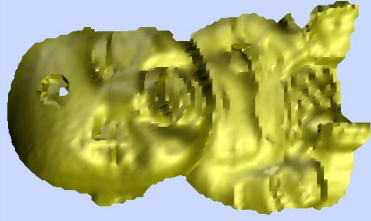
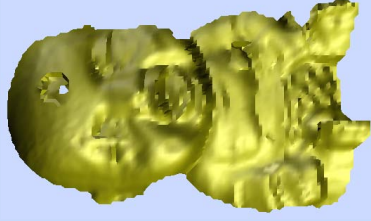
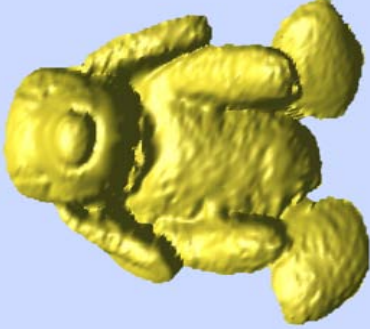
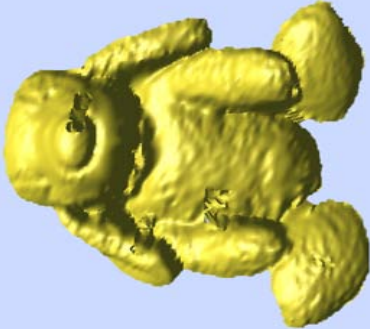
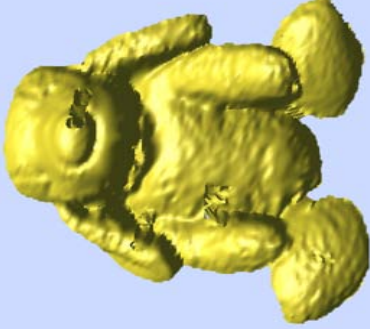
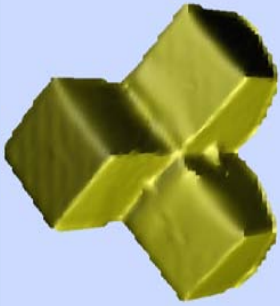
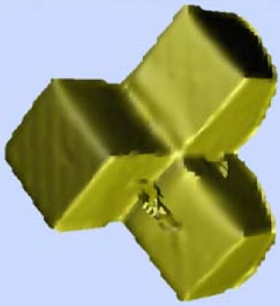
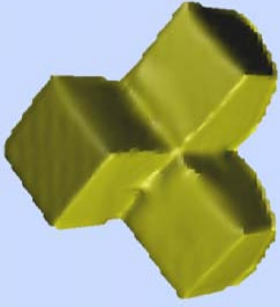
16			
17			
18			

Tabla 6.2: (continuación)

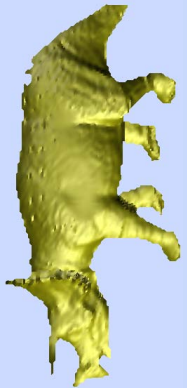
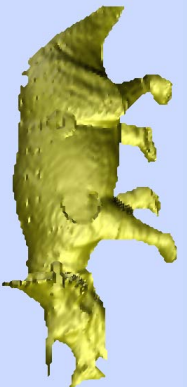
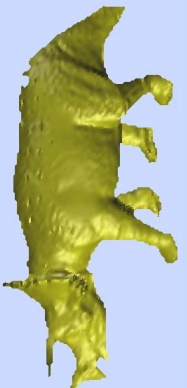
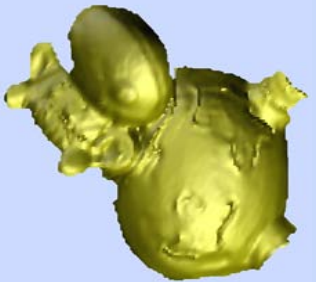
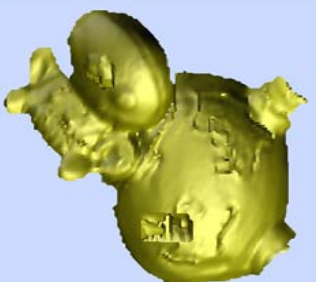
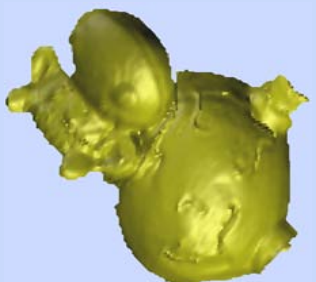
19			
20			

Tabla 6.3: Valor máximo y mínimo de la distancia entre nodos para los mapas de distancias

DISTANCIA ENTRE NODOS (mm)			
	Algoritmo 1	Algoritmo 2	Algoritmo 3
Valor máximo	4,097	33,092	4,761
Valor mínimo	0	0	0

Tabla 6.4: Mapas de distancias entre las mallas originales y las mallas reconstruidas

MAPAS DE DISTANCIAS DE LAS MALLAS RECONSTRUIDAS			
N Malla	Algoritmo 1	Algoritmo 2	Algoritmo 3
1			
2			
3			

Tabla 6.4: (continuación)

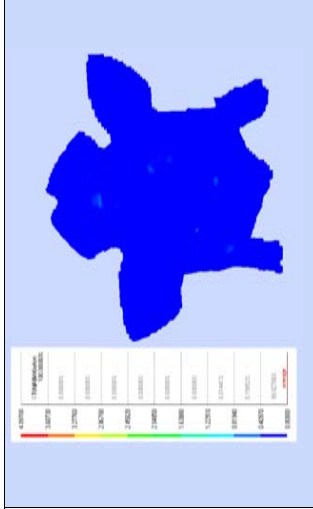
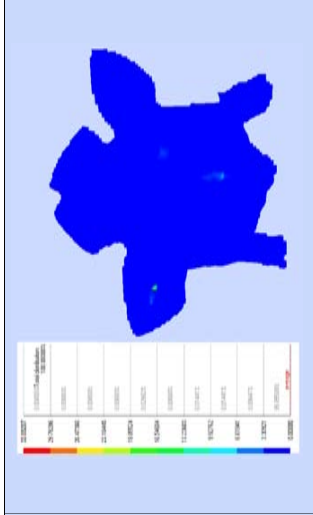
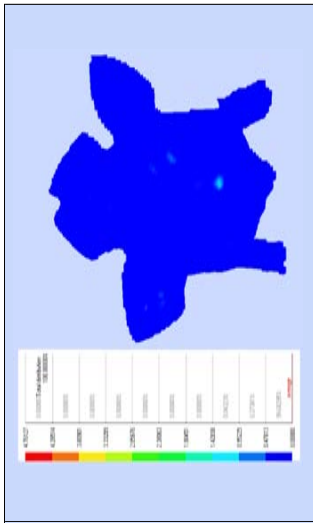
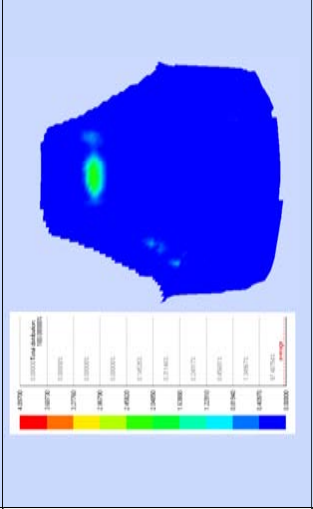
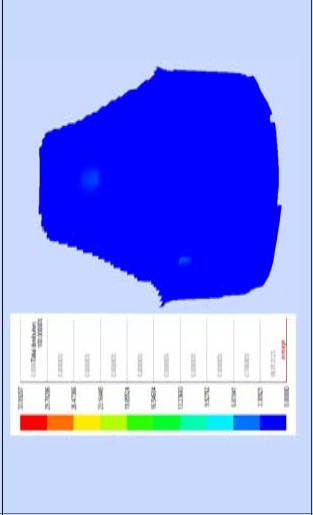
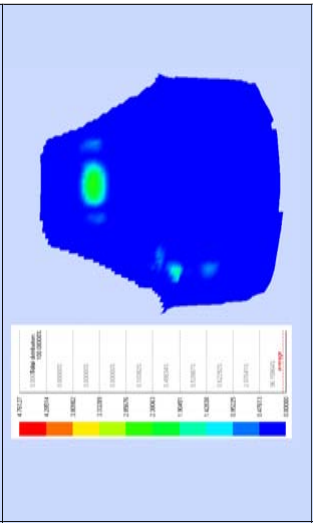
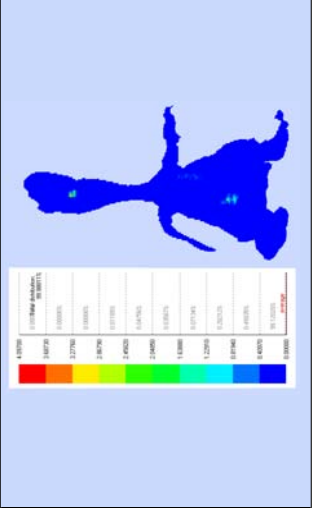
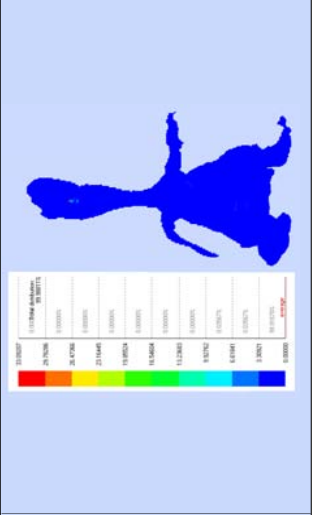
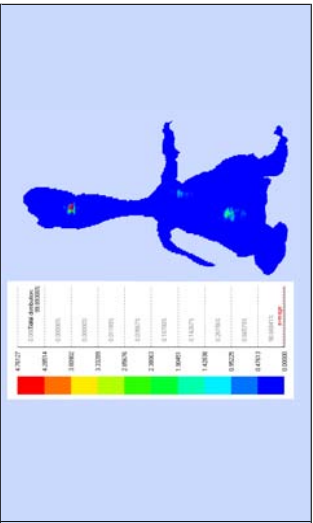
4			
5			
6			

Tabla 6.4: (continuación)

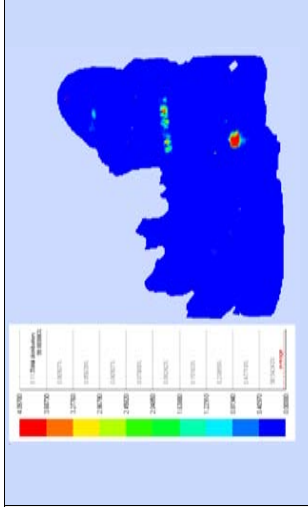
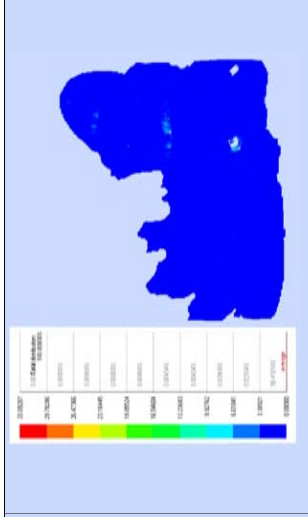
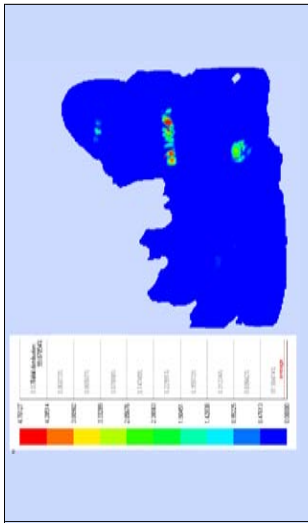
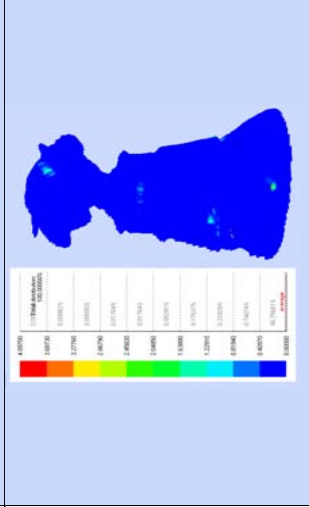
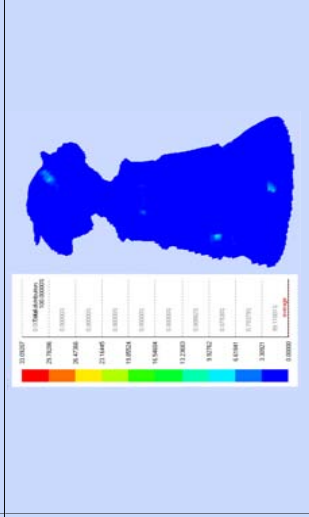
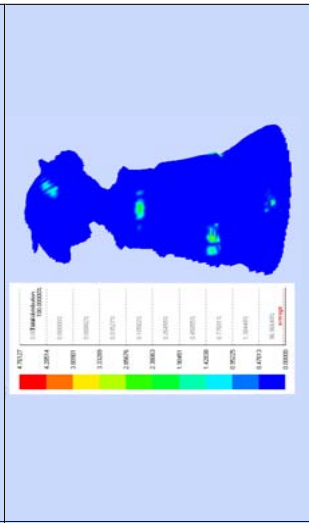
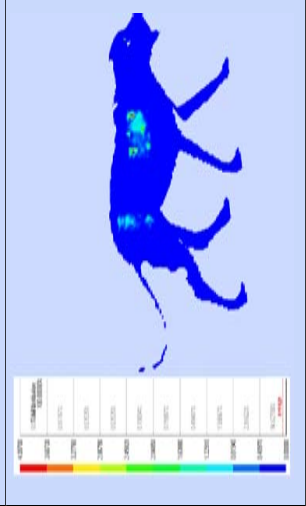
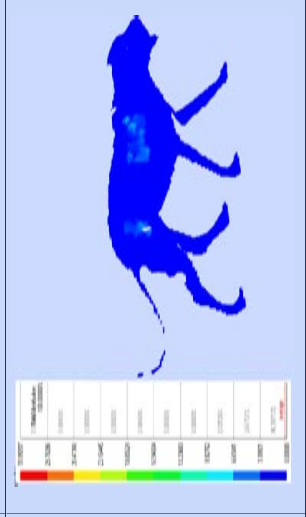
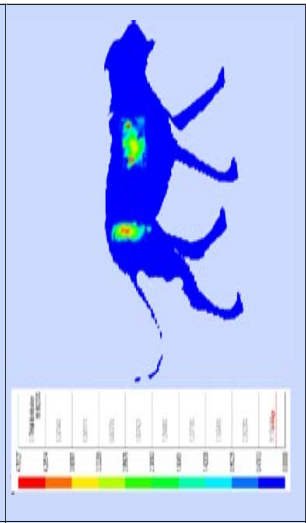
7			
8			
9			

Tabla 6.4: (continuación)

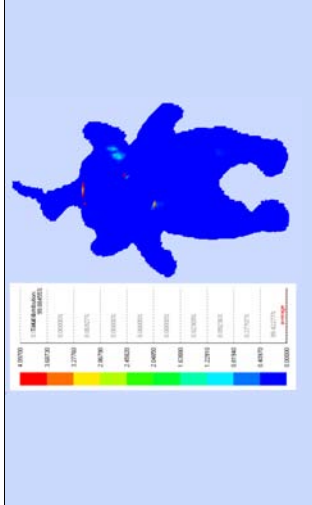
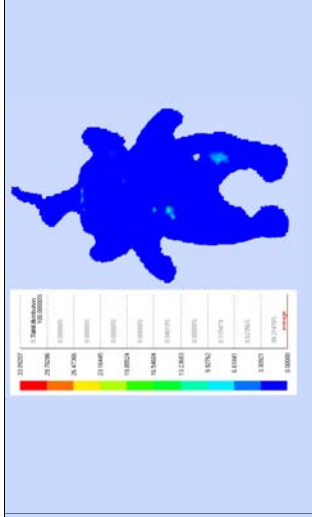
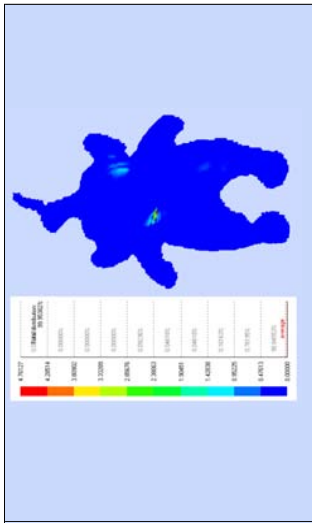
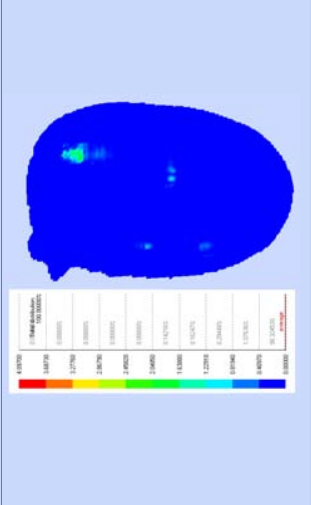
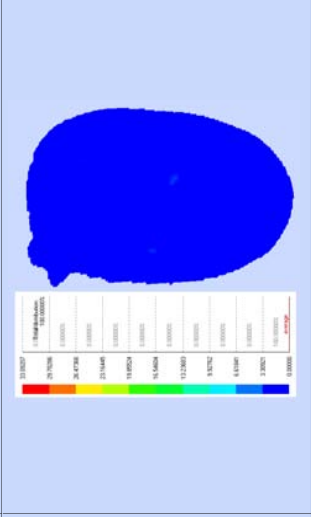
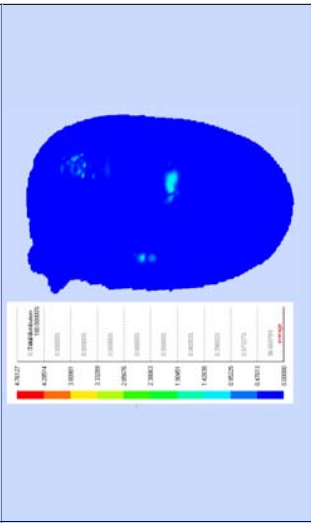
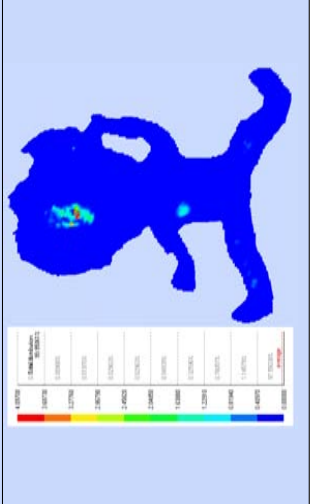
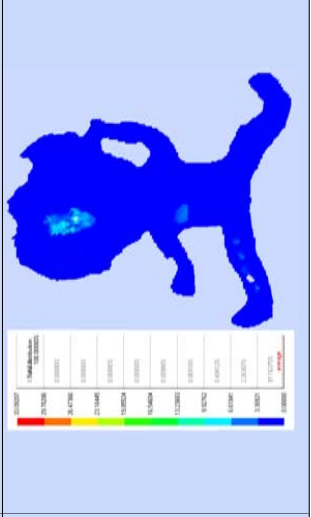
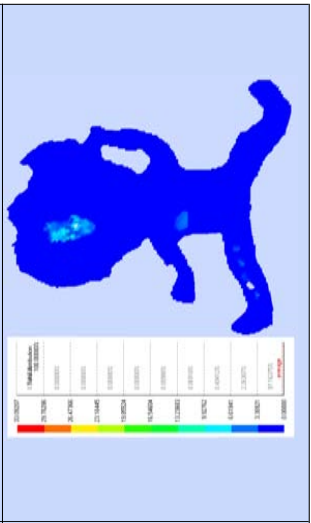
10			
11			
12			

Tabla 6.4: (continuación)

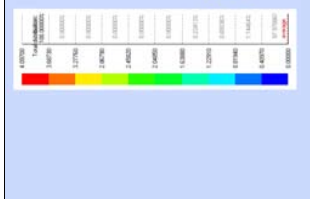
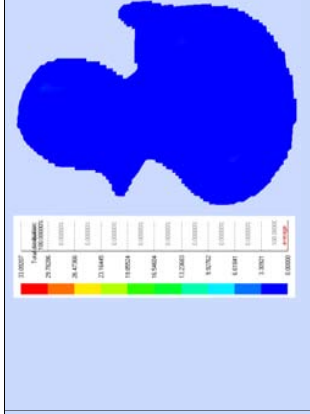
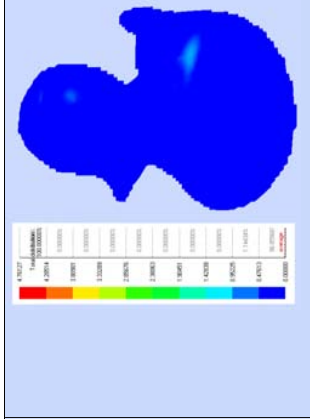
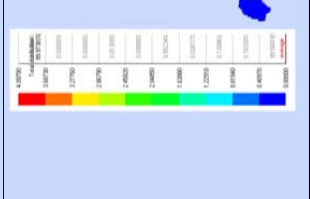
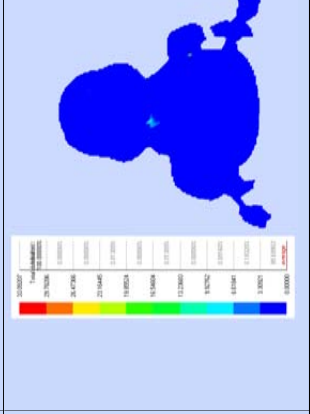
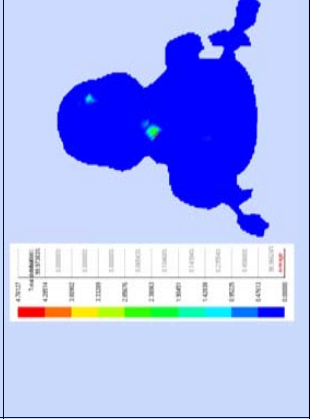

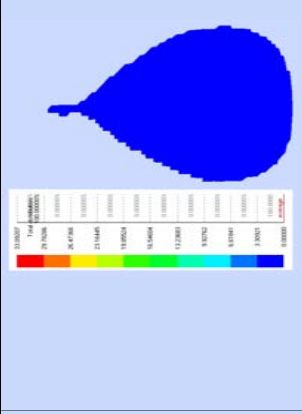
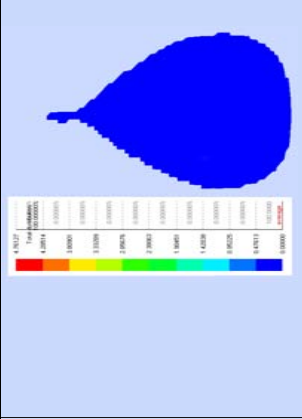
13			
14			
15			

Tabla 6.4: (continuación)

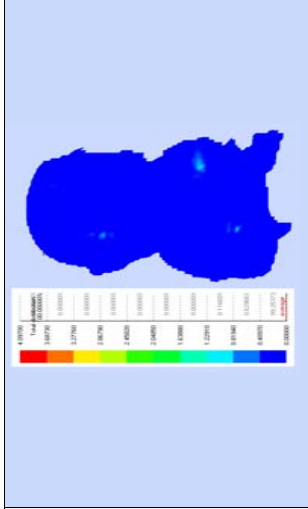

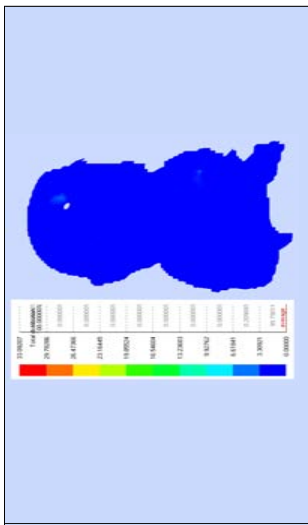
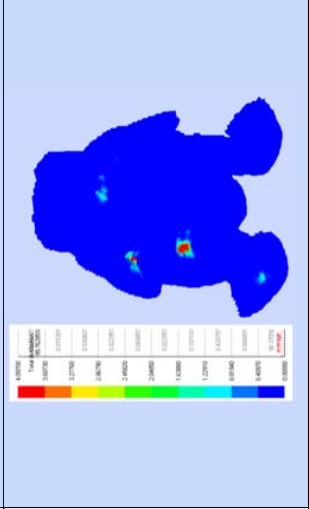
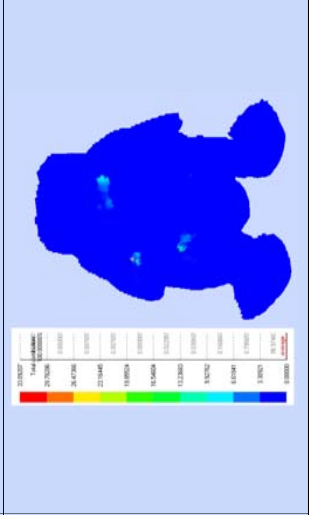
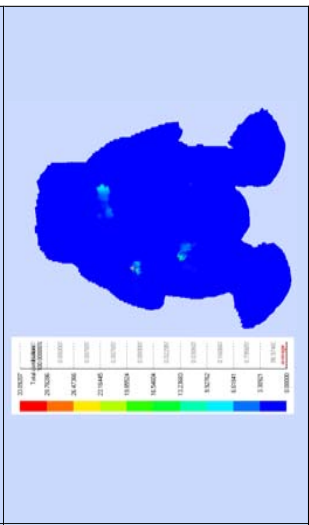
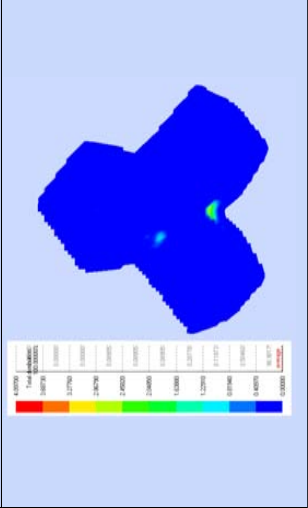
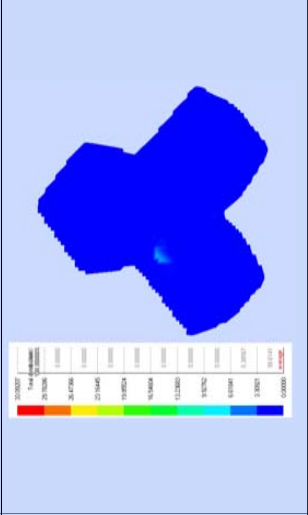
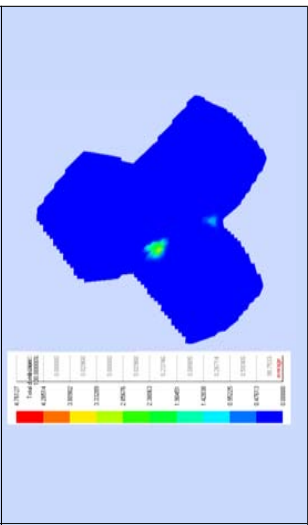
16			
17			
18			

Tabla 6.4: (continuación)

19			
20			

Capítulo 7

Anexo II

7.1. El equipo de captura.

7.1.1. Descripción específica del sensor de rango empleado.

En este proyecto el sensor de rango empleado para la obtención de las superficies tridimensionales de objetos es el Digitalizador de No contacto Vivid 300 fabricado por la casa Minolta. En la Figura 7.1 se muestran imágenes de dicho sensor así como un diagrama de las dimensiones del mismo.

El principio de medida de este digitalizador se basa en el empleo combinado de un emisor láser y un sensor CCD y la aplicación de técnicas de visión y triangulación a las imágenes adquiridas de la escena. Por tanto, no se requiere un contacto físico con la superficie del objeto, solo es necesario colocar el objeto delante del escáner en un entorno de iluminación adecuada.

En la Figura 7.2 se muestra un esquema detallado que ilustra el funcionamiento físico del sensor de rango. En esta se observa que el origen del sistema es un emisor láser que hace incidir un rayo de luz sobre una lente que produce la dispersión en un haz plano de luz. Este haz se refleja sobre un espejo giratorio (Galvano Mirror) que da lugar a un barrido del haz de luz sobre la superficie del objeto. Este proceso es captado por el sensor CCD que dispone el escáner y, mediante la aplicación de los algoritmos de cálculo de triangulación, se obtendrán como resultado las distancias de los puntos de la superficie al sensor.

Sobra decir que todo este proceso para la adquisición de una instantánea tridimensional del objeto es transparente al usuario, que sólo ha de operar en la colocación relativa del objeto y del sensor, en la iluminación adecuada de la escena y en un sencillo manejo del software específico suministrado.

Del modo de funcionamiento y estructura del escáner se deduce una ventaja que ofrece respecto a otros sensores del mercado: se trata de la portabilidad y movilidad del mismo que permite su utilización tanto en entornos cerrados como abiertos, así como la adquisición de grandes superficies sin más que ir recorriendo las mismas con el sensor. La superficie total se compondría a posteriori mediante las operaciones de registro suministradas en el software específico. En el caso de este proyecto, la utilización del escaner se reduce a un espacio cerrado con iluminación controlada y con posicionamiento y orientación del sensor mediante trípode.

Resaltar como curiosidad, de la lista de características enumeradas, los tiempos de operación de la máquina, que como se han detallado son de 0.6 segundos para el tiempo de escaneo y de 2 segundos o menos para el tiempo de transferencia de la imagen al ordenador,

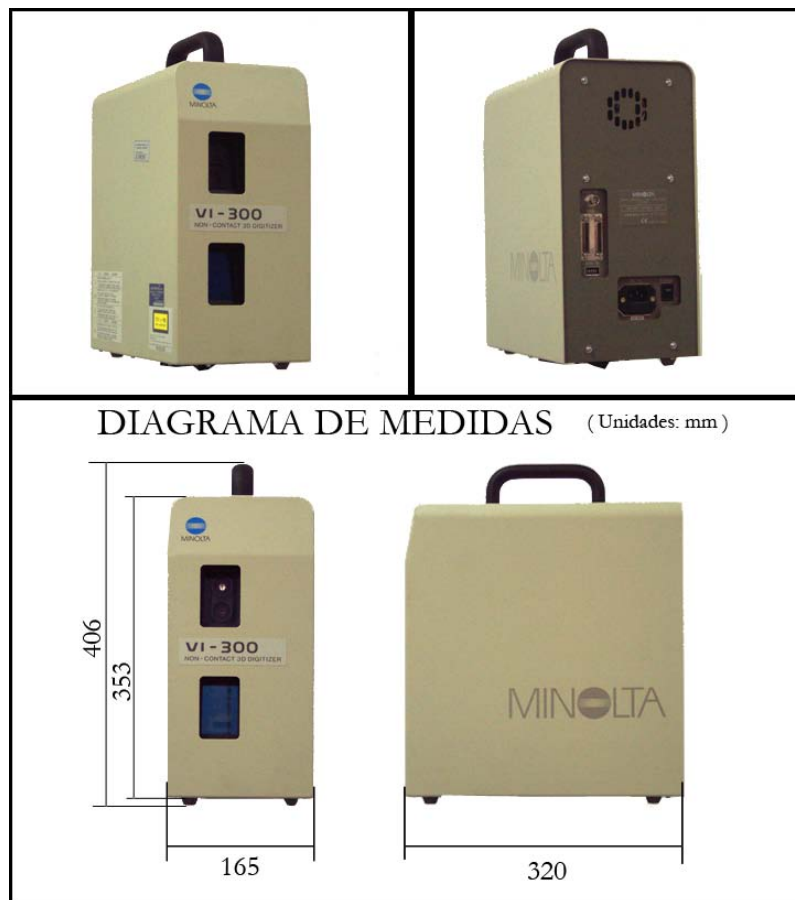


Figura 7.1: Vista frontal y trasera y diagrama de medidas del sensor Vivid 300 de Minolta.

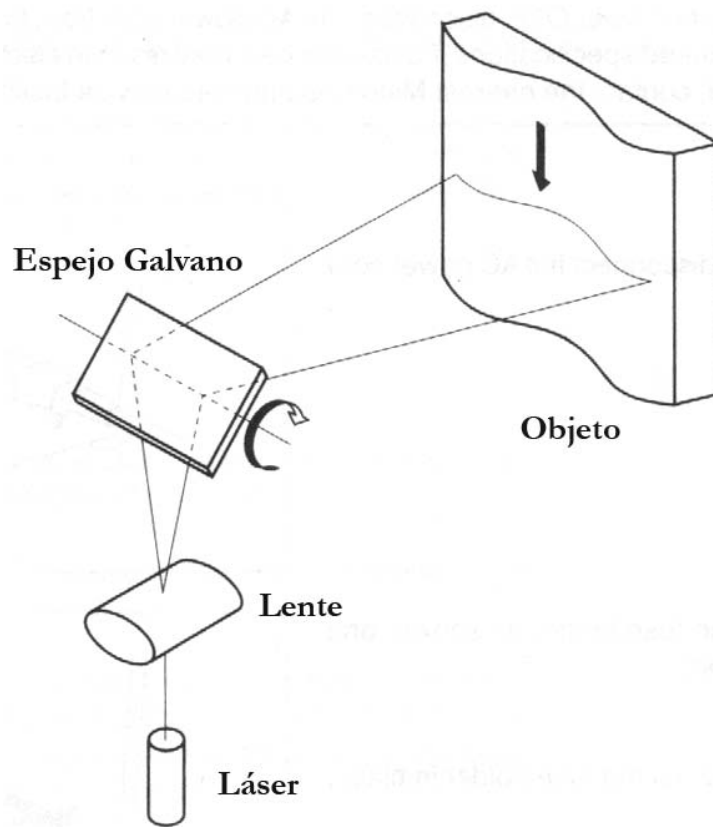


Figura 7.2: Principio de medida del sensor de rango.

tiempos que como se puede apreciar son bastante pequeños. Posteriormente se demostrará que estos tiempos resultarán insignificantes para el total de tiempo requerido para la obtención final de la malla del objeto, dado que la mayor parte de tiempo de dicho proceso la consumirán las operaciones de tratamiento de toda la información (mallas y texturas) aportada por el sensor de rango. Aunque también ha de resaltarse la importancia de una realización adecuada del escaneado, con una preparación adecuada del objeto y el entorno sobre el que se situará el mismo, debido a la incidencia que esto tiene sobre el tiempo total del proceso.

Seguidamente se enumeran algunas de las características técnicas de los elementos básicos que constituyen el sensor de rango.

- Modelo VIVID 300/VI-300
- Lente de recepción de la luz $f = 12 \text{ mm}$
- Potencia del láser $\lambda = 690 \text{ nm}$, Max. 7 mW(Class I o equivalente, controlado y emitido por el circuito de control interno y ópticas)
- Ángulo de propagación del haz Horizontal: 23.7° ; Vertical: 0.1°
- Método de escaneo láser Espejo Galvano
- Rango de distancia al objeto 0.55 a 1.2 m
- Campo de visión (cada lado del campo de visión xy) 185 a 395 mm
- Rango de distancias operativo (en z) Equivalente a cada lado del campo de visión (cuando la distancia de la imagen de entrada es 1.2m)
- Tiempo de escaneo 0.6 seg
- Tiempo de transferencia de datos de imagen al ordenador 2.0 seg o menos
- Condiciones de iluminación ambiente 500 lx o menos
- Controlador de imagen 3D data: CCD de $\frac{1}{2}$ pulgadas de transferencia de fotograma (380000 elementos de imagen)
- Color data: CCD color de $\frac{1}{2}$ pulgada (380000 elementos de imagen)
- Datos de salida Datos 3D: 200 x 200
- Datos de Color: 400 x 400
- Formato de salida Datos 3D: Formato original (convertidos a datos 3D por la utilidad software VI-S1)
- Datos de color: RGB 24-bit
- Interfaz de salida SCSI II, monitor color NTSC
- Características eléctricas 100 a 240 V \square (50 a 60 Hz), 0.6 A (medido a 100 V \square)
- Fusible Type T2A 250 V

- Dimensiones 165 (W) x 406 (H) x 320 (D) mm
- Peso 8 kg
- Rango de temperaturas de operación 10 a 35 ° C (85% de humedad relativa o menos, no condensación a 35 ° C)
- Rango de temperaturas de almacenamiento -20 a 50 ° C (85% de humedad relativa o menos, no condensación de 35 ° C)

A continuación seguirá una descripción detallada del resto de los elementos que conforman el equipo de trabajo así como de las conexiones y relaciones entre los mismos.

7.2. Descripción de los elementos y configuración empleados.

7.2.1. El trípode y la cabeza de rotación

Como soporte del escáner se dispone de un trípode profesional de aluminio modelo 475 del catálogo suministrado por la casa Manfrotto. Éste lleva adosada una cabeza de rotación, modelo 279 del catálogo Manfrotto, que facilitará enormemente las labores de colocación y orientación adecuadas del sensor según las necesidades impuestas por la escena a escanear.

En la Figura 7.3 se muestra una imagen del trípode sin accesorios adicionales, que servirá básicamente para ajustar la posición del escáner con respecto a la escena a capturar. Las principales características de éste son las que a continuación se enumeran:

- Columna central de cremallera con bloqueo
- Patas de aluminio adonizado
- Sistema de apoyo telescópico de medio nivel
- Manivela de acción rápida para el bloque de los extensores de las patas
- Topes de goma en las patas con pinchos retráctil
- Anillo para instalación de correa de transporte adicional
- Altura máxima alcanzable: 188 cm
- Peso del trípode: 4,3 kg
- Peso máximo soportado: 12 kg

Para controlar la orientación del escáner el trípode lleva adosado en la parte superior la cabeza de rotación o rótula que se muestra en la Figura 7.4. Ésta ofrece 3 ejes de rotación fácilmente ajustables manualmente, con la mejora añadida de incluir escalas graduadas y burbujas de nivel para cada eje que darán mayor precisión a la hora de fijar la orientación deseada. A continuación se enumeran algunas de las especificaciones de esta cabeza de rotación Manfrotto MN229:

- Máxima capacidad de carga: 11,9 kg



Figura 7.3: Imagen del trípode Manfrotto 475 utilizado como soporte del escáner.

- Altura de la cabeza: 16 cm
- Giro: 360 °
- Inclinación: -25° +90° inclinación frontal, -90° +30° inclinación lateral
- Peso de la cabeza: 1,9 kg

Es evidente que este elemento no es ni mucho menos imprescindible para la adquisición del tipo de objetos con que se ha trabajado en este proyecto pero hay que incidir en que facilita enormemente las labores de posicionamiento y orientación del escaner y que nos permite tomar instantáneas de la escena desde enfoques que sin el trípode no serían posibles. En la Figura 7.5 se muestra una imagen de cómo queda el conjunto montado: escáner, cabeza de rotación y trípode.

7.2.2. La mesa o tabla de rotación. El patrón de calibración.

Como accesorio para el proceso de escaneado completo de un objeto, se dispone de una mesa de rotación, que se muestra en la Figura 7.6. Ésta consiste básicamente en un plato giratorio accionado por un motor paso a paso, que produce la rotación del objeto a intervalos de grados determinados, para así ir obteniendo vistas del objeto espaciadas uniformemente a lo largo de un giro de 360°.

Esta mesa de rotación está fabricada con materiales metálicos robustos de un tono negro mate para, de esta manera, interferir lo menos posible en las condiciones de iluminación de la escena a escanear. El plato giratorio tiene un diámetro de 20 cm lo que da una idea del tamaño de objetos que podrán colocarse sobre la misma.

La mesa de rotación habrá de conectarse por un lado a la red eléctrica para alimentar el motor y los circuitos auxiliares y por otro lado al controlador del plato giratorio de la casa ISEL (Figura 7.7). Este es un elemento que traduce recibe las órdenes del ordenador a través de una conexión por puerto serie y va generando las señales para el giro del plato.



Figura 7.4: Imagen de la cabeza de rotación Manfrotto 279 utilizada para la orientación del escáner.



Figura 7.5: Imagen del conjunto formado por trípode, la cabeza de rotación y el escáner.

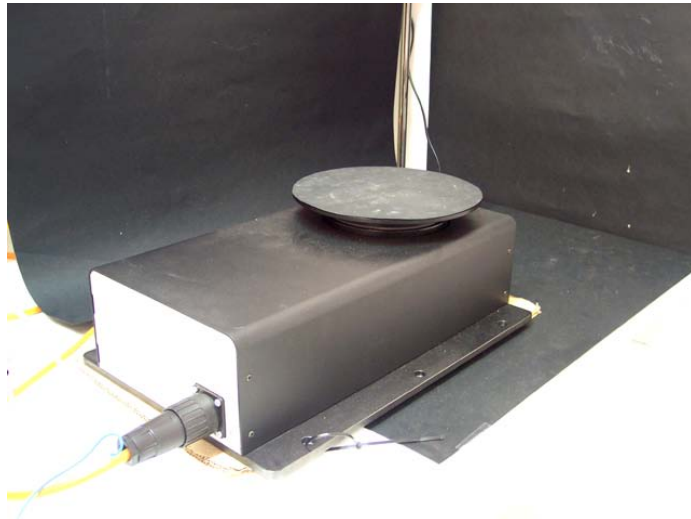


Figura 7.6: Imagen de la mesa de rotación situada en el entorno de captura con iluminación controlada

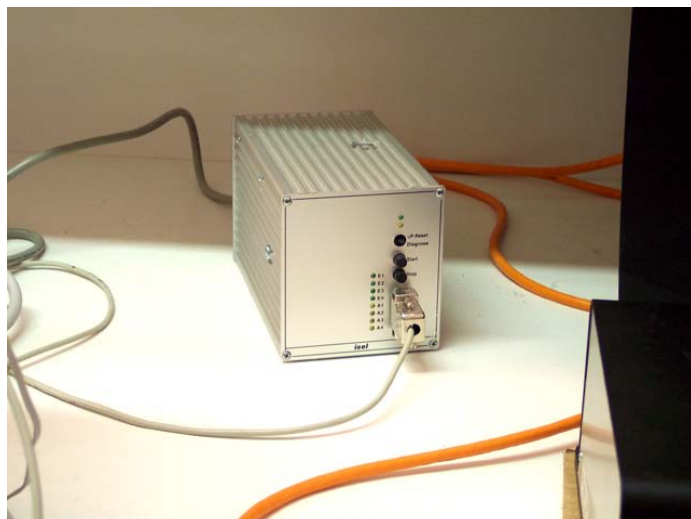


Figura 7.7: Controlador de la mesa de rotación de la marca ISEL

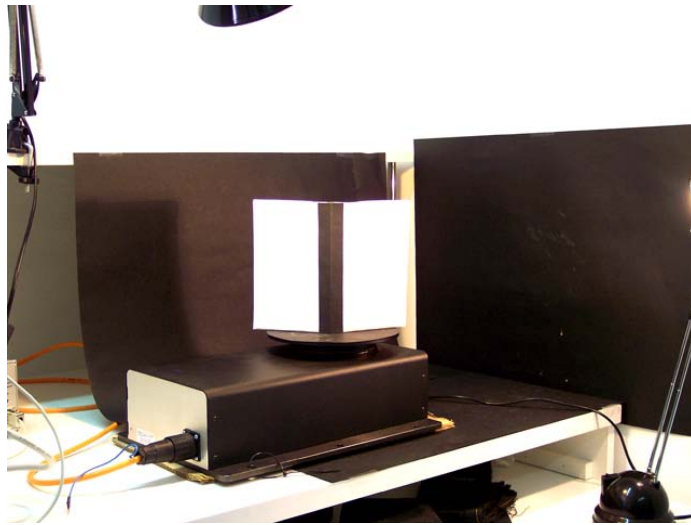


Figura 7.8: Imagen del patrón de calibración centrado sobre el plato giratorio

Es evidente que este accesorio presenta la limitación de sólo poder trabajar con objetos de un cierto tamaño y peso. Las piezas con que se trabaja en este proyecto tienen unas dimensiones y peso que en ningún caso se aproximan a los límites máximos que soportan los elementos del plato giratorio.

El manejo del plato se hará desde el software específico proporcionado con el escáner mediante un controlador específico que también es suministrado con el software. Desde el programa hay accesible una opción para realizar el escaneado utilizando la mesa de rotación. El usuario deberá introducir el ángulo entre vistas que desea, y el software procederá a ir controlando la rotación del plato y a ir mostrando los resultados para cada instantánea de escáner. Un paso imprescindible una vez finalizado el giro completo del objeto, es la colocación del patrón de calibración, una vez que es solicitada por el software. Este patrón, que se muestra en la Figura 7.8, consiste básicamente en un elemento fabricado en plástico que está formado por dos planos opacos de color blanco que forman un ángulo entre sí de 120° . En la zona convexa de la unión entre los dos planos hay una franja de 5 cm de espesor de color negro. La función de este patrón no es otra que señalar al software la situación del eje de rotación del plato, situación que es diferente cada vez que cambia la posición y orientación relativas del escáner y mesa de rotación. Por tanto, el usuario tendrá que, una vez concluida la rotación completa del objeto y tras ser solicitado por el software, colocar el patrón de calibración de forma que la recta de intersección entre los dos planos coincida aproximadamente con el eje de rotación del plato. Para ello, este plato dispone de unas guías en las que encajar los pivotes salientes que tiene el patrón en su base de apoyo.

7.2.3. La iluminación de la escena.

El ajuste de la iluminación de la escena es un paso importante en el proceso de escaneado de un objeto. Una mala iluminación puede dar lugar a un aumento en el tiempo de conformación de la malla completa así como a una pérdida de calidad tanto en las mallas como en las texturas resultantes. Para el caso concreto de este proyecto se han utilizado como accesorios de iluminación dos flexos articulados de luz blanca y de no muy elevada potencia luminosa, que pueden observarse en la Figura 7.9. El entorno donde se sitúan los objetos que se desean escanear se ha preparado en tonos oscuros con dos finalidades principalmente: la



Figura 7.9: Flexos utilizados para el control de iluminación de la escena.



Figura 7.10: Entorno de captura con iluminación controlada

primera, conseguir controlar mejor la iluminación del objeto y la segunda, facilitar las labores de registro y mezcla posterior de las texturas de cada vista. También antes se mencionó que la superficie de la mesa de rotación era de color negro por los mismos motivos que se han señalado. En la Figura 7.10 se muestra una imagen del entorno con iluminación controlada donde se situarán los objetos para la adquisición de superficies.

A pesar de esta iluminación cuidada que se ha procurado, realmente el sensor de rango con que estamos trabajando no presenta excesivos problemas en la adquisición de superficies bajo condiciones de iluminación pobres, sobre todo en relación con otros sensores que sí presentan una gran sensibilidad ante este parámetro. Aunque, como se ha dicho anteriormente, sí es verdad que un control de estas condiciones da lugar a una mayor calidad para cada vista del objeto, tanto en las mallas como en las texturas resultantes y, de igual manera, a una reducción del tiempo total necesario para completar la malla de un objeto.

Un parámetro de las superficies de los objetos que está muy relacionado con la iluminación de la escena es el brillo que presentan. Así, una superficie excesivamente brillante dará

problemas en el proceso de escaneado, ya que el haz de láser reflejado por ella será menos visible por el sensor CCD. Esto es debido a que se produce una saturación lumínica en las zonas especulares de la superficie. Este problema da lugar a que las mallas resultantes del proceso de adquisición presenten huecos o agujeros en esas zonas donde se producía un mayor reflejo de la luz. Por tanto, se ha de procurar elegir objetos cuya superficie no tenga excesivo brillo. Aun así, si fuese necesario escanear un objeto con tales características superficiales, que plantease dificultades en la adquisición de su superficie, por los problemas anteriormente comentados, se puede optar por la aplicación de una pintura mate a la superficie del objeto. Esta opción ha sido llevada a cabo con algunos objetos de la base de datos y se le ha aplicado una capa de pintura acrílica de color crema mate. Tras este tratamiento de la superficie se hace patente una mejora sustancial en los resultados de adquisición de superficie, y se consigue, por tanto, una reducción de tiempo total del proceso. Es evidente que el gran inconveniente que presenta esta opción es la pérdida de la textura real del objeto que es ocultada por la capa de pintura.

Como antes se mencionó, la iluminación de la escena influye enormemente en la calidad de la textura que se obtendrá tras la adquisición. Esto viene limitado lógicamente por las características físicas de la cámara del escáner que fueron enumeradas. En ésta se observaba el parámetro Condiciones de iluminación ambiente que presentaba un valor de 500 lx o menos. Por tanto, habrá que fijar las condiciones de iluminación de forma que no sean demasiado pobres, y den lugar a texturas con elevado ruido, ni excesivamente elevadas, dando lugar a zonas de saturación lumínica en la textura. Se perseguirá pues que resulte una textura rica en información del color real del objeto.

7.2.4. El monitor de video.

Este es un elemento que, aunque no es imprescindible en el proceso de escaneado, facilita enormemente las labores de posicionamiento y orientación relativas del escáner y de la escena.

En el caso concreto de este proyecto el monitor elegido es un Sony PVM-14L1, como el mostrado en la Figura 7.11, cuyas características principales se enumeran en la Tabla 2. Este monitor dispone de una entrada de video NTSC que es el formato de salida de video del sensor Minolta. Se ha optado por un monitor en color que se utilizará como apoyo para las siguientes funciones: posicionamiento y orientación del escáner y la escena; control de la iluminación, colocación del patrón de calibración.

Como se puede apreciar en la tabla de características, el monitor dispone de 2 líneas de entrada video, de las que lógicamente solo ocuparemos una. De igual manera ofrece multitud de controles configurables para ajustar los parámetros de la imagen en los niveles deseados. En la Figura 7.12 se muestran dos imágenes del monitor, donde se pueden apreciar los controles accesibles en el frontal y las salidas disponibles en la parte trasera del mismo.

A pesar de la ayuda que nos proporciona este elemento, es necesario resaltar un inconveniente que presenta y que habrá que tener en cuenta en la utilización habitual del mismo. Este inconveniente no es otro que la discrepancia entre la imagen mostrada por el monitor y la imagen sobre la que se aplica el proceso de triangulación, que es la que es visible desde el software específico del escáner. En la Figura 7.13, se muestra un ejemplo de tal discrepancia, que pone de manifiesto el margen de error a tener en cuenta al utilizar el monitor de video como guía para las funciones antes mencionadas.

Pese a este ligero inconveniente, se ha resaltar el gran apoyo que nos brinda el monitor y la consiguiente reducción de tiempo a que da lugar el empleo del mismo.



Figura 7.11: Monitor de video Sony PVM-14L1 utilizado para visualizar la señal de video del escaner

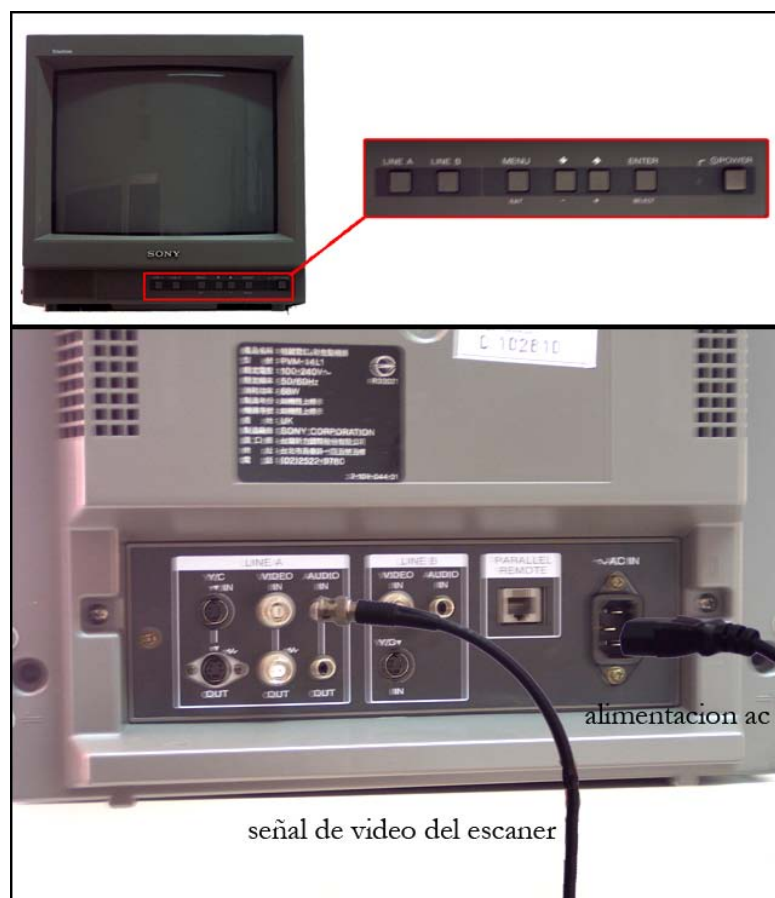


Figura 7.12: Controles accesibles en el frontal del monitor y conexiones de entrada y salida disponibles en la parte trasera del mismo

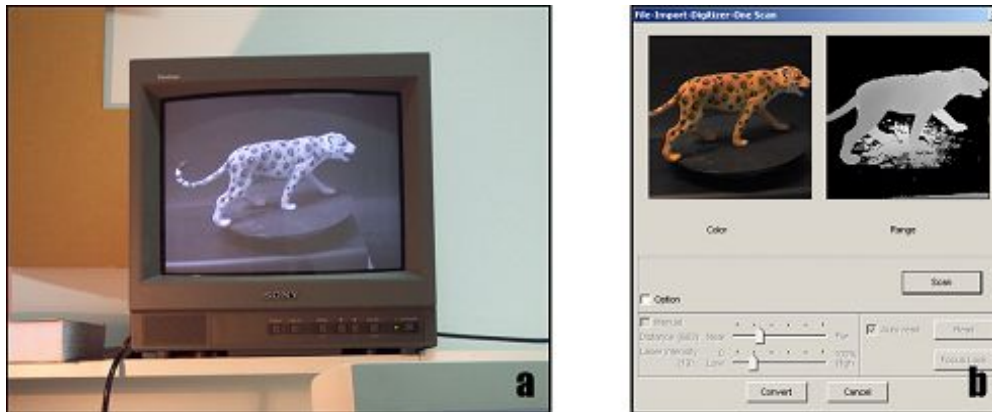


Figura 7.13: (a). Imagen captada por la cámara del escáner que es mostrada en el monitor de video;(b). Imagen a la que realmente se le aplica el proceso de triangulación

7.2.5. El ordenador.

Hardware.

Este es el elemento principal del equipo de escaneado, desde el que se llevan a cabo las operaciones de control del escáner, del plato giratorio, tratamiento de mallas y texturas,... Todo esto se realiza por medio de un software específico suministrado por el fabricante del sensor.

En el ordenador es necesario instalar una tarjeta SCSI para la comunicación con el escáner. Se ha elegido que esta interconexión se lleve a cabo mediante SCSI debido al elevado flujo de información entre el ordenador y el sensor que debe ser capaz de gestionar.

Por otra parte, el plato giratorio se comunica con el ordenador mediante un puerto de comunicaciones serie. A diferencia del caso anterior, la comunicación entre estos dos elementos no conlleva el trasiego de una gran cantidad de información por lo que bastará con la velocidad que nos ofrece la comunicación serie.

En cuanto a las características del hardware sobra decir que, para las tareas que debe realizar el ordenador en todo el proceso, se requiere elevada potencia de cálculo así como elevada cantidad de memoria y gran capacidad de almacenamiento, debido al gran volumen de datos que se ha de gestionar. En el caso concreto de este proyecto el equipo empleado es un Pentium IV 2 GHz, 1 GB RAM, 40 GB HDD, tarjeta gráfica ATI Radeon 9600 Pro 256 MB DDR; se muestra una imagen del mismo en la Figura 7.14.

Interconexiones entre elementos.

Como resumen se muestra en la Figura 21 un esquema general de relaciones entre los diferentes elementos que componen el sistema. De esta forma se observa que el elemento central de control es el ordenador. A éste se conecta el escáner, por medio de un enlace SCSI, y la mesa de rotación con su correspondiente controlador, que lo hacen por medio de un enlace RS-232.

Por otra lado se puede establecer una conexión lógica entre el ordenador y el software específico.

Por su parte, el escáner estaría conectado con una conexión física a la cabeza de rotación que a su vez tiene una conexión física con el trípode.

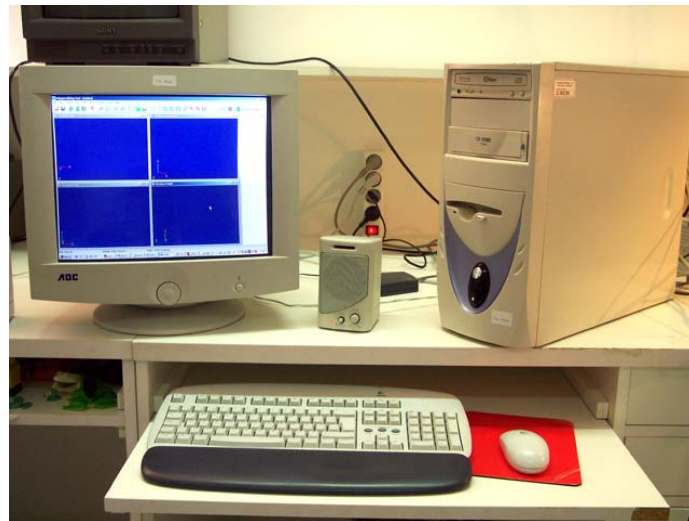


Figura 7.14: Ordenador que se ha utilizado para el sistema de adquisición de superficies

Por último quedaría una conexión básica a la red eléctrica de los siguientes elementos: escaner, mesa de rotación con su controlador y ordenador.

7.3. Estudio del brillo del objeto

Otro aspecto importante a tener en cuenta antes de acometer la fase de adquisición será la cantidad de brillo que presente la superficie del objeto. Si la superficie del objeto es totalmente mate, en general, no habrá ningún problema en cuanto a la adquisición de la escena ante cualquier configuración de la iluminación. En cambio, en cuanto presente una mínima cantidad de brillo habrá que comprobar la incidencia que éste tenga en la fase de adquisición y la problemática a que puede dar lugar su presencia para las distintas vistas. Para ello se efectuarán diversas pruebas de escaneado de diversas vistas haciendo las pertinentes variaciones de las condiciones de iluminación. Se estudiarán por tanto las configuraciones de la iluminación de la escena que sean más idóneas para la adquisición de la superficie, es decir, aquellas en las que las pérdidas de información ocasionadas por el brillo sean mínimas. Por estas pérdidas se entienden las discontinuidades en la malla o agujeros que se presentan en las zonas donde se alcanza una gran intensidad de brillo.

De igual manera, se notarán los efectos negativos en la textura, que presentará zonas de saturación de luz en los puntos de máximo brillo, lo cual se traduce en la pérdida de la textura en dichas zonas. En el caso de superficies con un grado muy elevado de brillo, que presenten grandes problemas en la fase de adquisición, se estudiará la posibilidad de aplicar una capa de pintura mate que elimine los brillos de la superficie. Con esto se tiene solucionado el problema ocasionado por el excesivo brillo pero da lugar lógicamente a una pérdida de la textura del objeto. Una solución a este inconveniente sería la reconstrucción manual de la textura por medio de fotografías del objeto, tomadas antes de la aplicación de la capa de pintura. Se trataría de un proceso manual y poco exacto que dependería de la pericia del usuario con el trabajo de mezclado de las fotografías. En la Figura 32 se muestra un ejemplo en el que se observa la influencia que el brillo tiene en el proceso de adquisición. De igual manera se muestra en la Figura 33 las diferencias que resultan al adquirir un objeto antes y después de la aplicación de una capa de pintura.

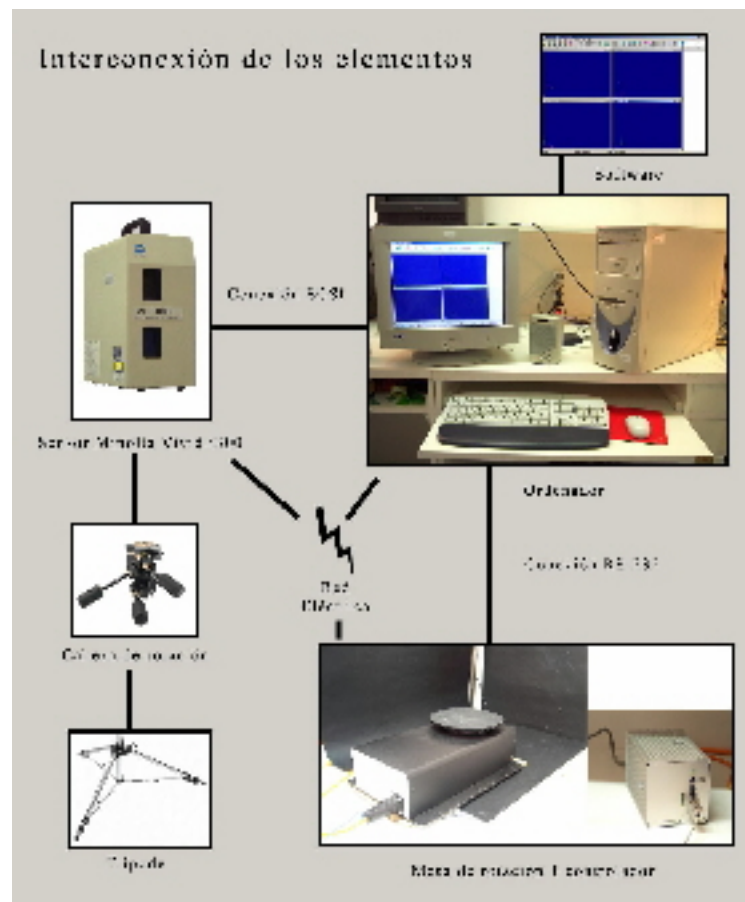


Figura 7.15: Esquema general que muestra las interconexiones entre los elementos que componen el sistema

7.4. Estudio de la textura. Problemas ocasionados por los tonos oscuros.

Otro problema que se puede presentar en el proceso de escaneado viene derivado de los tonos que presente la textura del objeto. Como se explicó en el punto 1.2 del Capítulo I, el sensor de rango obtiene las profundidades de la escena a partir de visión del haz de láser recorriendo la superficie del objeto, pues bien, para el caso de superficies con zonas de tonos muy oscuros el escáner presenta problemas en la adquisición. Estos problemas serán mayores lógicamente cuanto mayor sea la extensión de la zona oscura respecto a la superficie total del objeto. En las Figuras 34, 35 y 36 se pueden observar algunos ejemplos de captura de objetos con zonas negras de varios tamaños. De ellas también se deduce que el efecto negativo de tener una tonalidad oscura puede sumarse al de presentar un brillo excesivo, con el consiguiente empeoramiento de la calidad de la malla resultante. En cualquier caso habría que hacer las pertinentes pruebas de adquisición y decidir entre diversas opciones posibles: proceder al escaneado del objeto, si las zonas oscuras tienen un tamaño pequeño, reconstruyendo posteriormente mediante el software los agujeros a que darían lugar en la malla; aplicar una capa de pintura para eliminar esas zonas oscuras, con la consiguiente pérdida de la textura del objeto; en caso extremo, desechar el objeto para el escaneado.

7.5. Detección y eliminación de puntos y caras erróneas.

Un aspecto muy importante a tener en cuenta tanto en el tratamiento de la malla como en la finalización de la misma, es la detección y eliminación de los puntos y las caras erróneas. Esta tarea convendrá realizarla cada cierto tiempo tanto si los errores son evidentes como si no.

Una de las herramientas más susceptible de generar datos de superficie erróneos es la de rellenado de huecos de forma automática. Por lo que una vez ejecutada habrá que testear si se han producido errores. Si se da el caso de que los errores son repetitivos para un mismo hueco habrá que estudiar la aplicación de otro método de rellenado o la edición de los bordes de la malla en tal hueco.

7.5.1. Tipos de corrección de errores aplicados:.

- Find Abnormal Faces: Realiza una búsqueda de caras que cumplen una serie de características que se consideran anormales. Una vez ejecutada aparece un cuadro de diálogo como el de la Figura 112 en el que se selecciona qué tipo de caras anormales se desea seleccionar de entre los siguientes:
- Non-manifold faces: Este tipo de anomalía se refiere a la situación en que las caras no están “bien repartidas”. Éstas pueden causar muchos errores en varios cálculos propios del proceso de tratamiento, que son muy significativos en el registro de mallas. Geométricamente se trata de la situación en que varias caras comparten 3 o más aristas como puede verse en la Figura 7.16 donde las caras (V1 V2 V4), (V2 V4 V3) y (V2 V4 V5) comparten la arista (V2 V4).
- Redundant faces: Esta anomalía se produce desde el punto de vista de los vértices. Exceptuando los vértices del borde de malla, un vértice debe tener un número idéntico

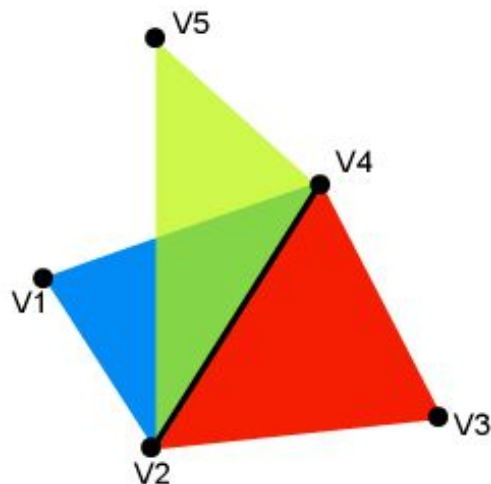


Figura 7.16: Ejemplo de caras con topología Non-manifold.

de caras y aristas incidentes. Si existe una cara que viole tal regla se considera una cara redundante. Un ejemplo de esta situación se muestra en la Figura 7.17 donde la cara (V1 V2 V3) es redundante.

- Crossing faces: Cuando dos aristas que comparten un vértice intersectan entre sí se da una situación geoméricamente incorrecta que puede causar muchos errores de cálculo. Este tipo de caras se denominan caras cruzadas y se puede ver un ejemplo de las mismas en la Figura 7.18 en donde la arista (V2 V3) de la cara (V1 V2 V3) está cruzada con la arista (V1 V4).
- Unstable faces: En este caso se realiza una comprobación de las caras que no comparten todas sus aristas con otra cara, es decir que de alguna manera son bordes de la malla. De esta forma pueden darse dos tipos de fallos:
 - Normales erróneas: en el que un grupo de caras presentan una normal diferente a la del entorno donde se ubican. En la Figura 7.19 se muestra un ejemplo de este tipo de fallos.
- ● Twisted Faces o caras giradas: en el que se generan un grupo de caras giradas en una especie de espiral, como el ejemplo de la Figura 7.20.

7.5.2. Operaciones a realizar para corrección de la malla:

- Clean Non-manifold Faces: Borrado de las caras “mal repartidas” que han sido explicadas en el apartado anterior.
- Clean Crossing Faces: Borrado de caras cruzadas explicadas en el apartado anterior.
- Find Self Intersection: Borrado de autointersecciones de la malla. En esta situación, de la que se muestra en ejemplo en la Figura 7.21, dos caras que no son adyacentes intersectan entre ellas.

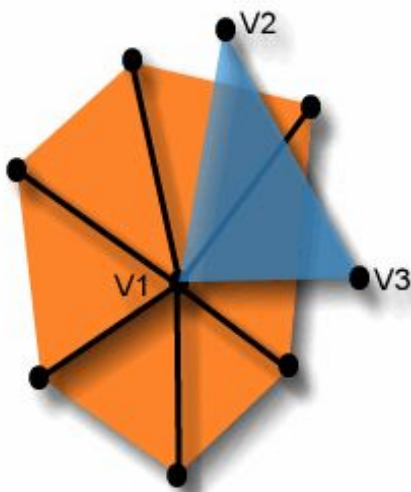


Figura 7.17: Ejemplo en donde la cara $(V1 V2 V3)$ es una cara redundante.

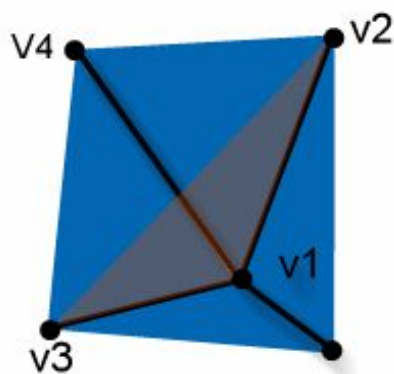


Figura 7.18: Ejemplo de caras cruzada en donde la arista $(V2 V3)$ de la cara $(V1 V2 V3)$ está cruzada con la arista $(V1 V4)$

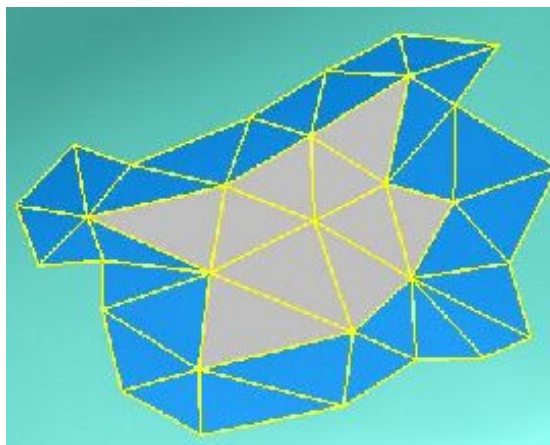


Figura 7.19: Ejemplo de caras inestables en la modalidad de normales erróneas

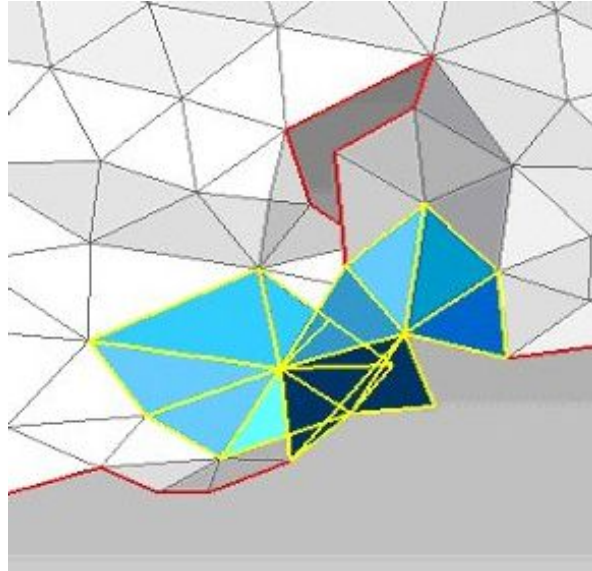


Figura 7.20: Ejemplo de caras inestables en la modalidad de caras giradas



Figura 7.21: Ejemplo de dos caras que intersectan entre sí.

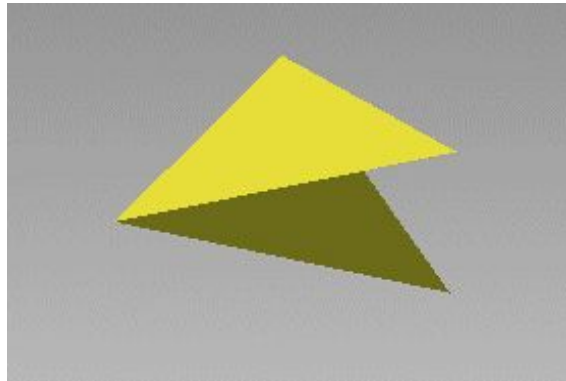


Figura 7.22: Ejemplo de “folded faces”, es decir, caras adyacentes que comparten una arista y con un ángulo entre ellas muy pequeño.

- Delete Spikes: Este comando busca caras que crean un pico en la superficie y permite al usuario seleccionarla o eliminarlas.
- Fix Bad Normals: Lleva a cabo la reparación de normales invertidas. Al ejecutarlo se presentan dos opciones excluyentes:
 - Topología consistente: Se marca en el caso de que la malla no tenga caras “mal repartidas” o caras non-manifold.
 - Geometría consistente: Se marca esta opción si la malla presenta problemas topológicos. En este caso se asume que la malla debe estar cerrada.
- Delete Singularities: Este comando lleva a cabo la eliminación de singularidades de la malla, tales como:
 - Vertices with no face: vértices sueltos que no corresponden a ninguna cara.
 - Coincident Vertices: vértices coincidentes.
 - Folded faces: aquellas caras que comparten una arista y que el ángulo entre ellas es muy pequeño. En la Figura 7.22 se muestra un ejemplo de este tipo de caras.
- - Faces on boundary: caras que tienen 2 ó 3, según se marque, aristas de borde.
 - Small clusters: grupos de caras interconectadas a través de un vértice que tienen un número de caras marcado por el cuadro de texto.
 - Zero area faces: caras cuyo área es inferior al valor marcado en el cuadro de texto.
 - Blank group: grupos sin ninguna cara.
 - Blank shell: mallas sin ningún vértice.
- Sew Boundary (Cosido de bordes): En ocasiones los vértices se posicionan formando una línea estrecha como en un agujero. Mediante este comando se proyectan los vértices de borde sobre la arista de borde más cercana y se añade al borde. En el cuadro de texto que se abre se introducirá la tolerancia que se le permite a la distancia proyectada que se ha explicado. En la Figura 7.23 se muestra el estado antes y después de ejecutar el cosido de bordes.

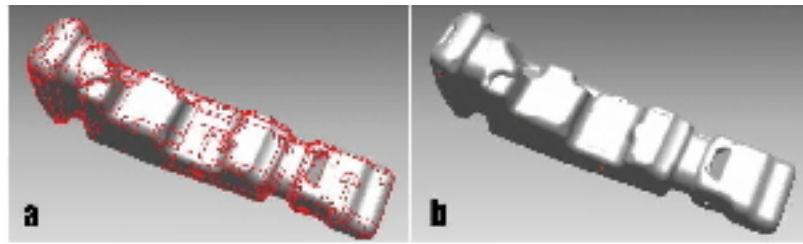


Figura 7.23: Figura 123.- (a) y (b) Estado anterior y posterior, respectivamente, al cosido de bordes.

7.6. Estudio de la ocupación óptima

Tabla 7.1: Estudio de la ocupacion optima

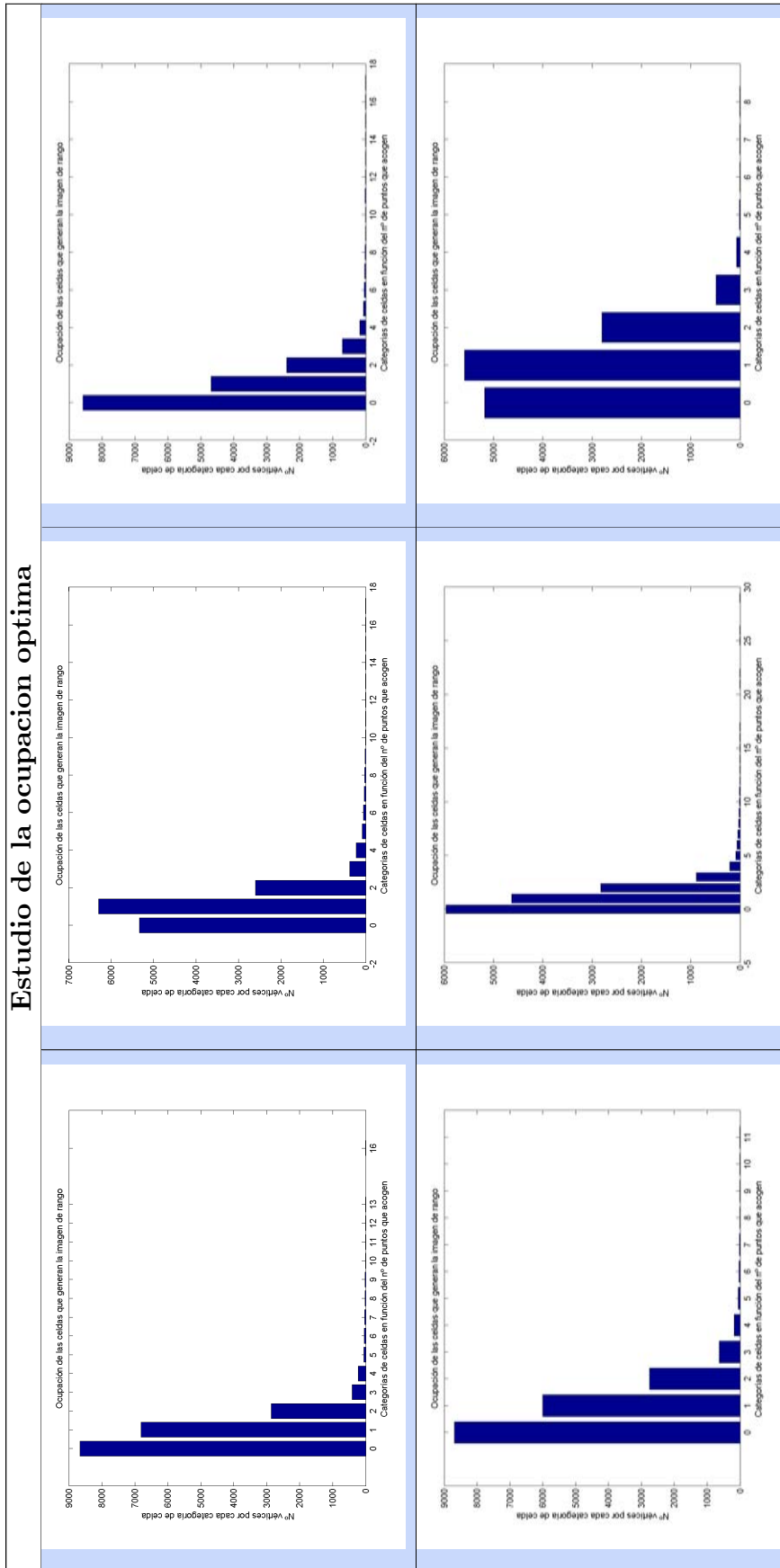


Tabla 7.1: (continuación)

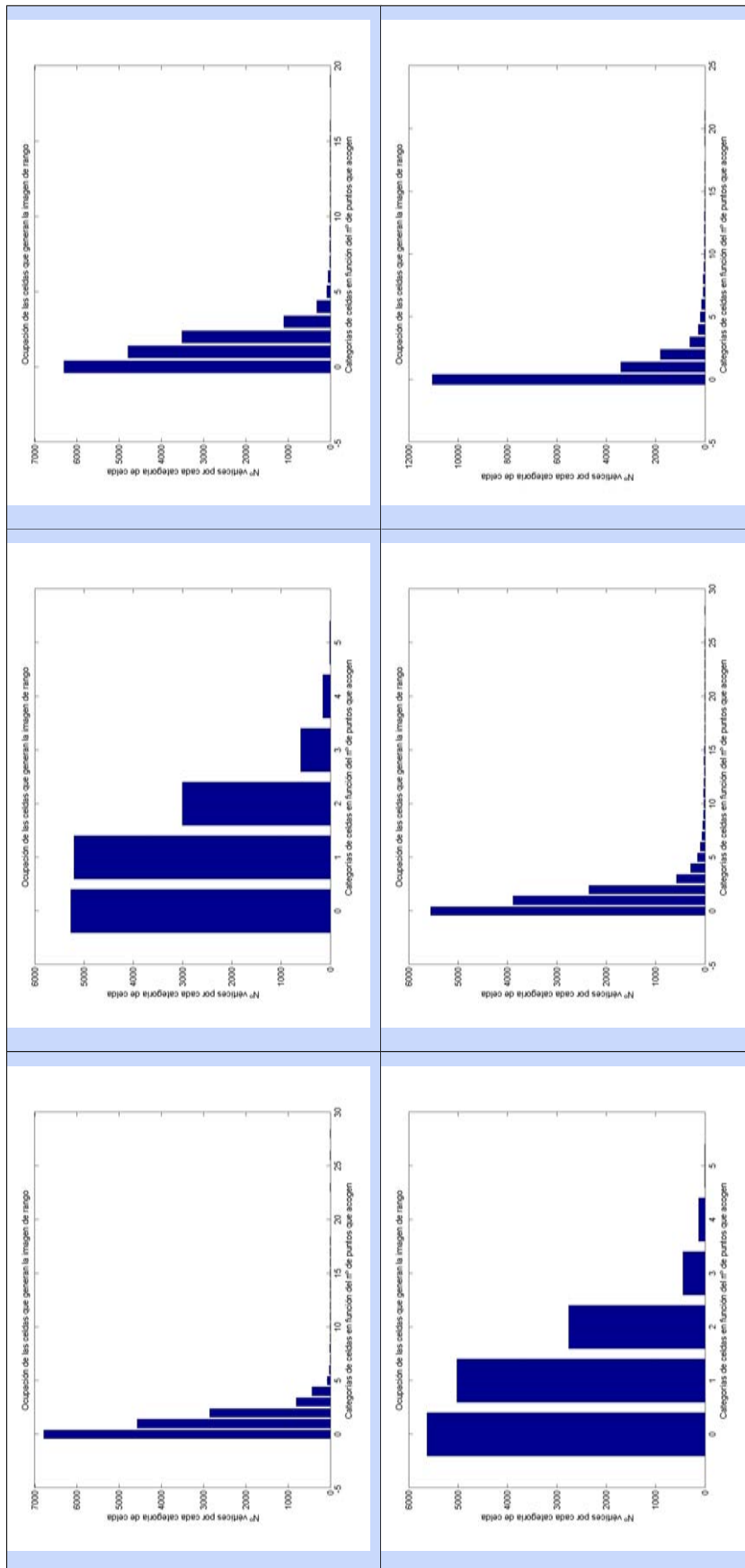


Tabla 7.1: (continuación)

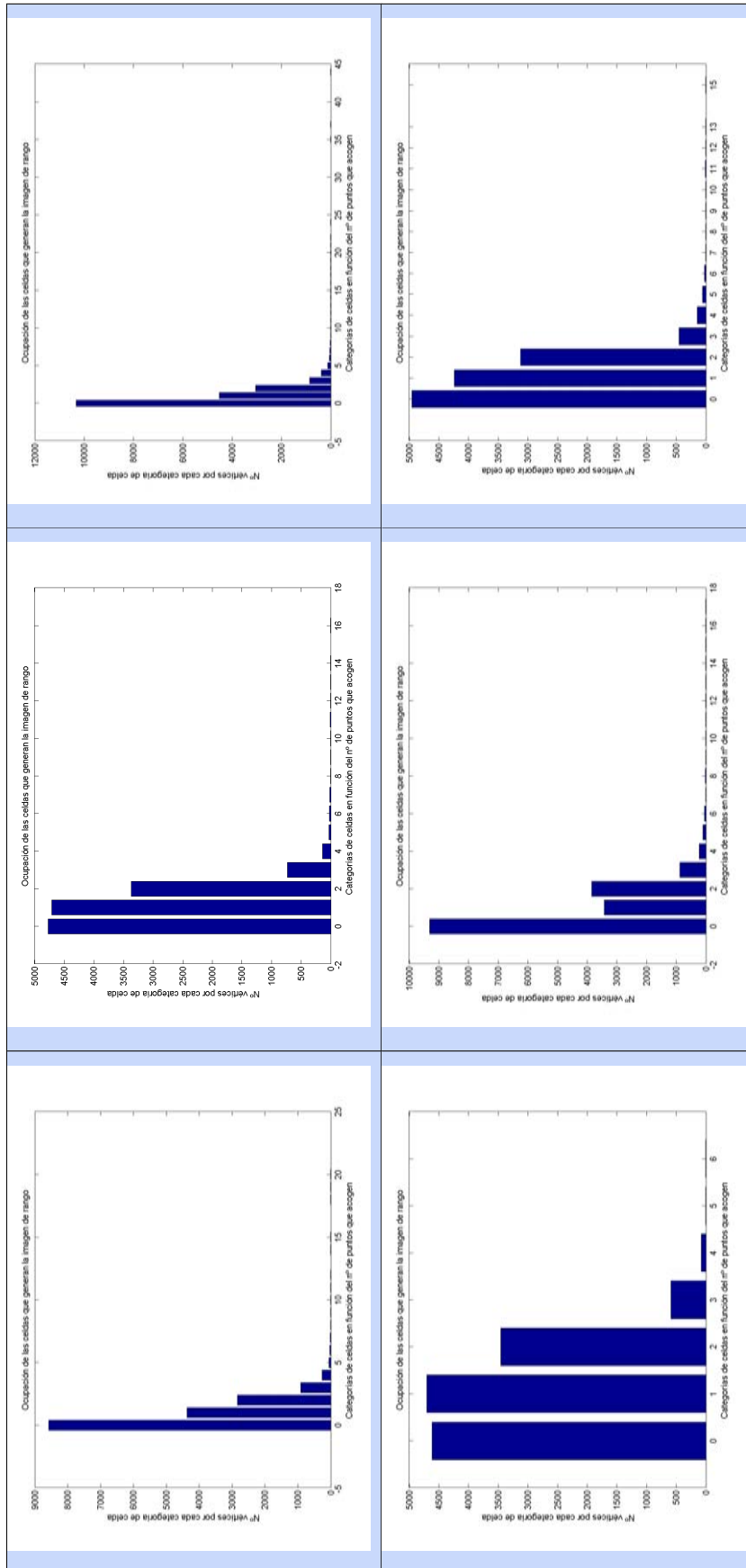


Tabla 7.1: (continuación)

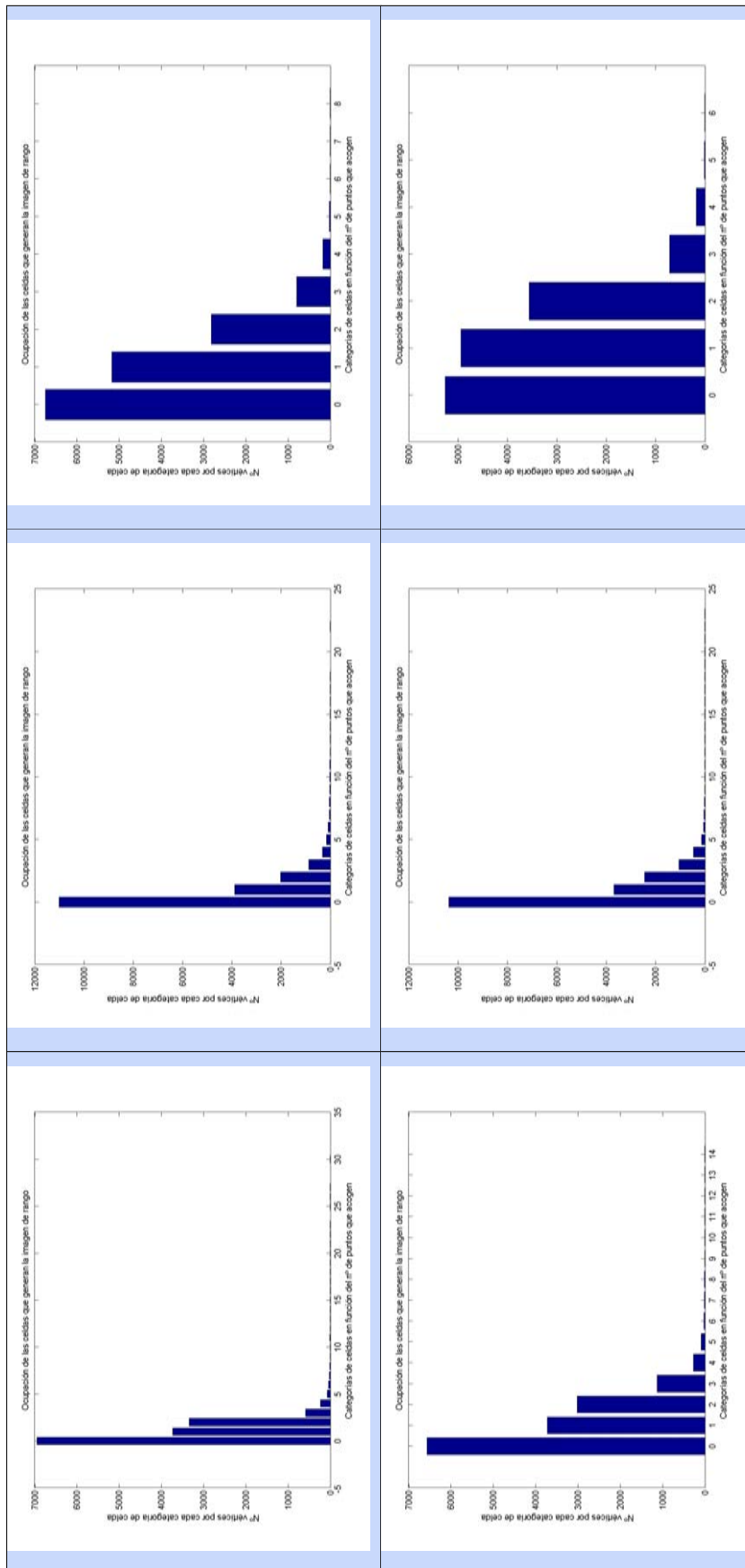


Tabla 7.1: (continuación)

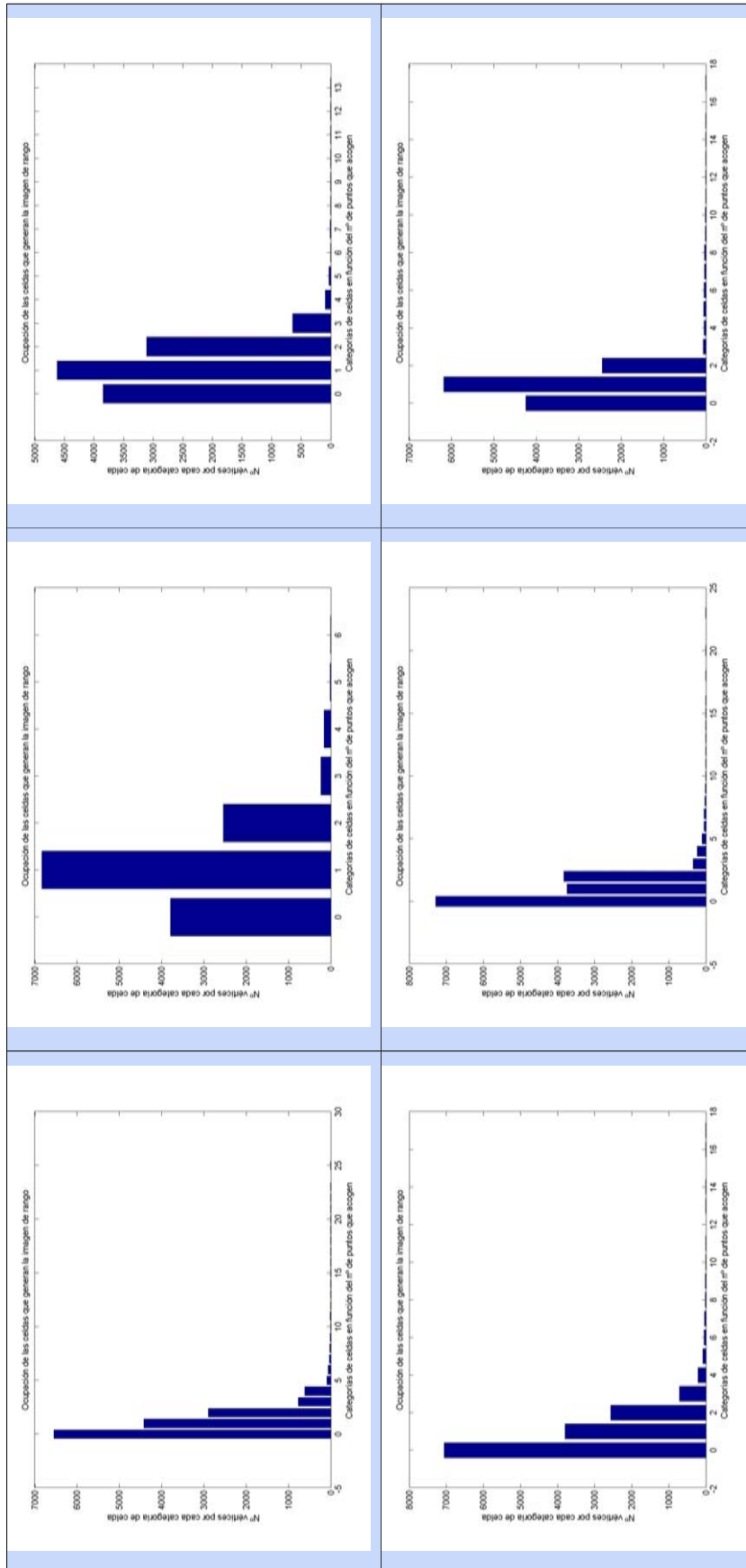


Tabla 7.1: (continuación)

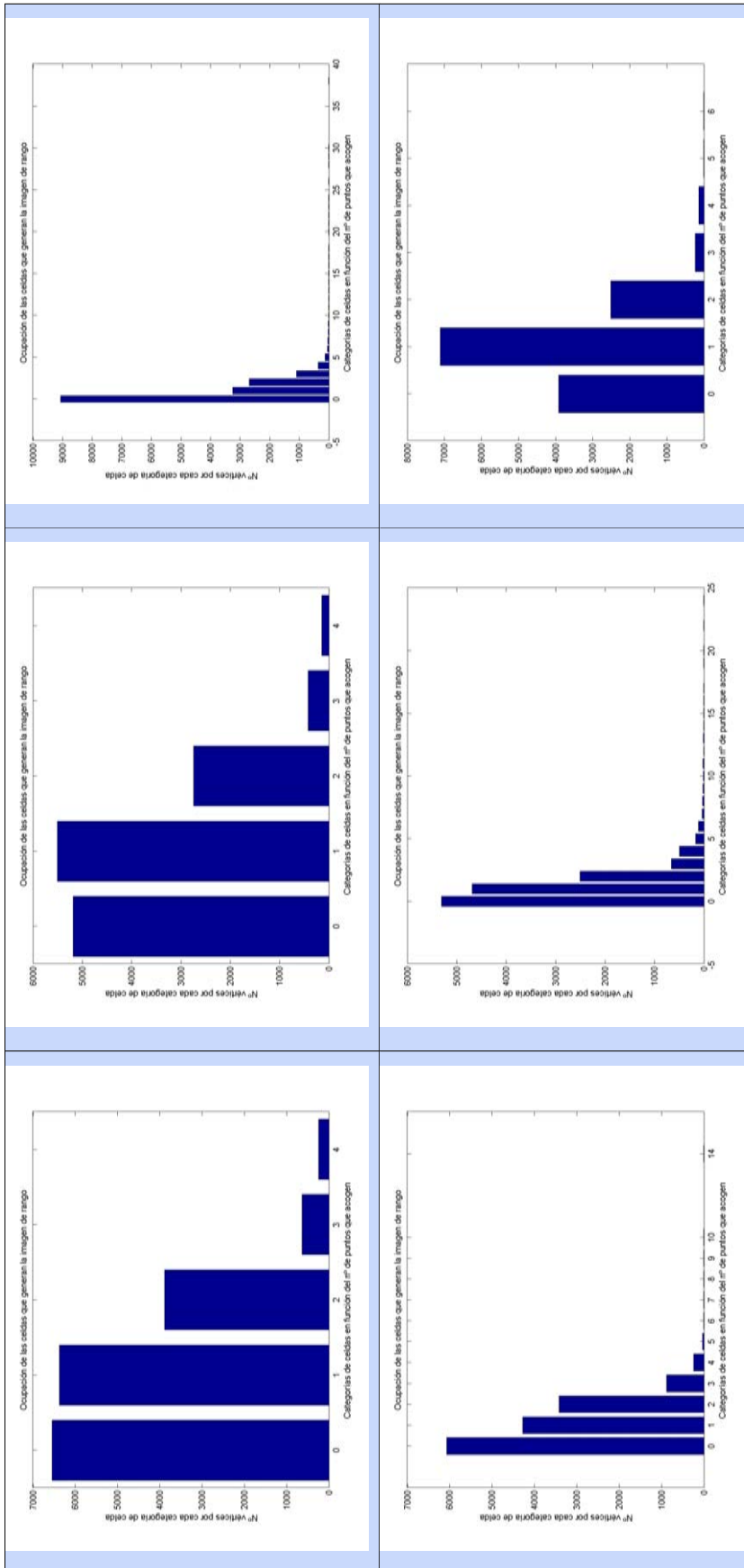


Tabla 7.1: (continuación)

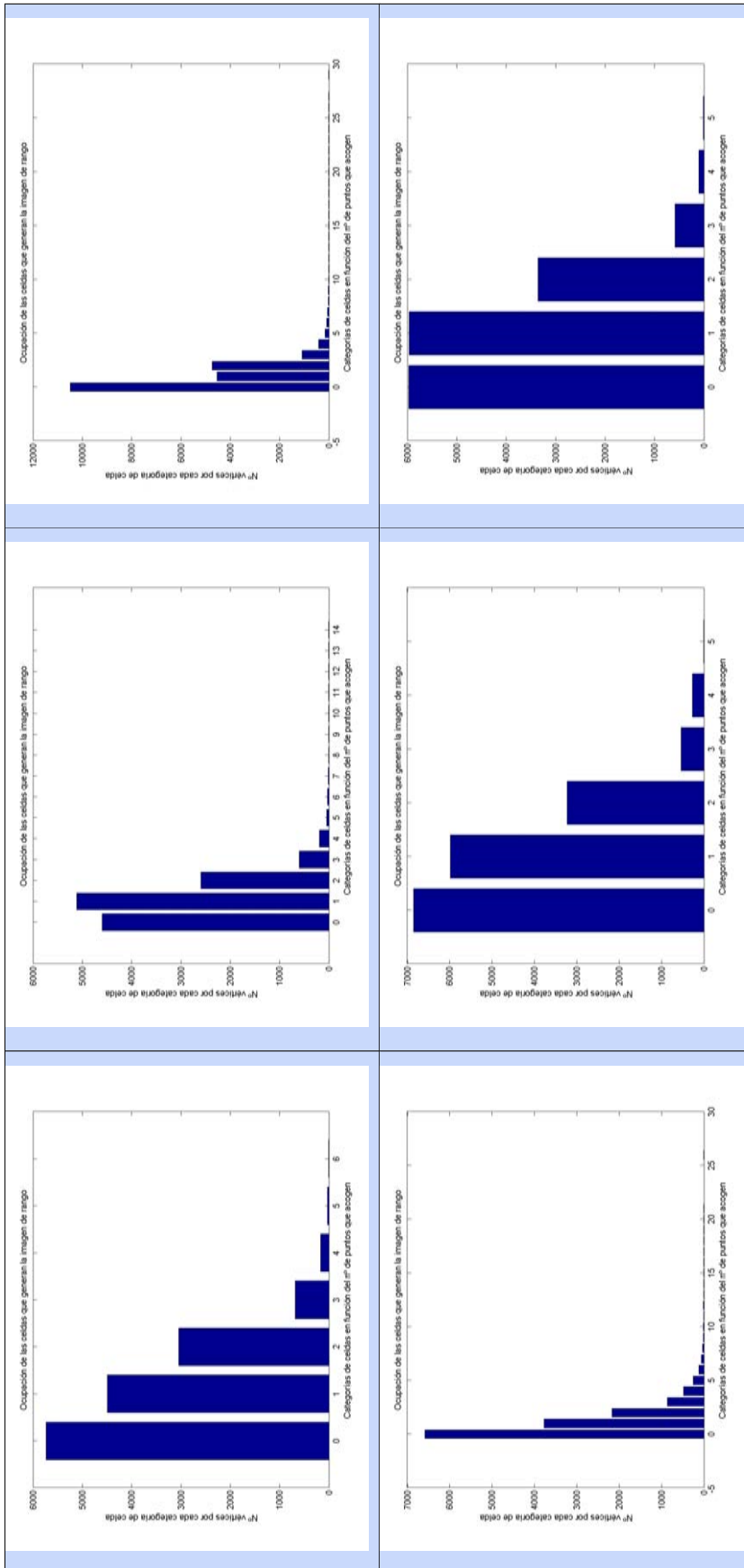


Tabla 7.1: (continuación)

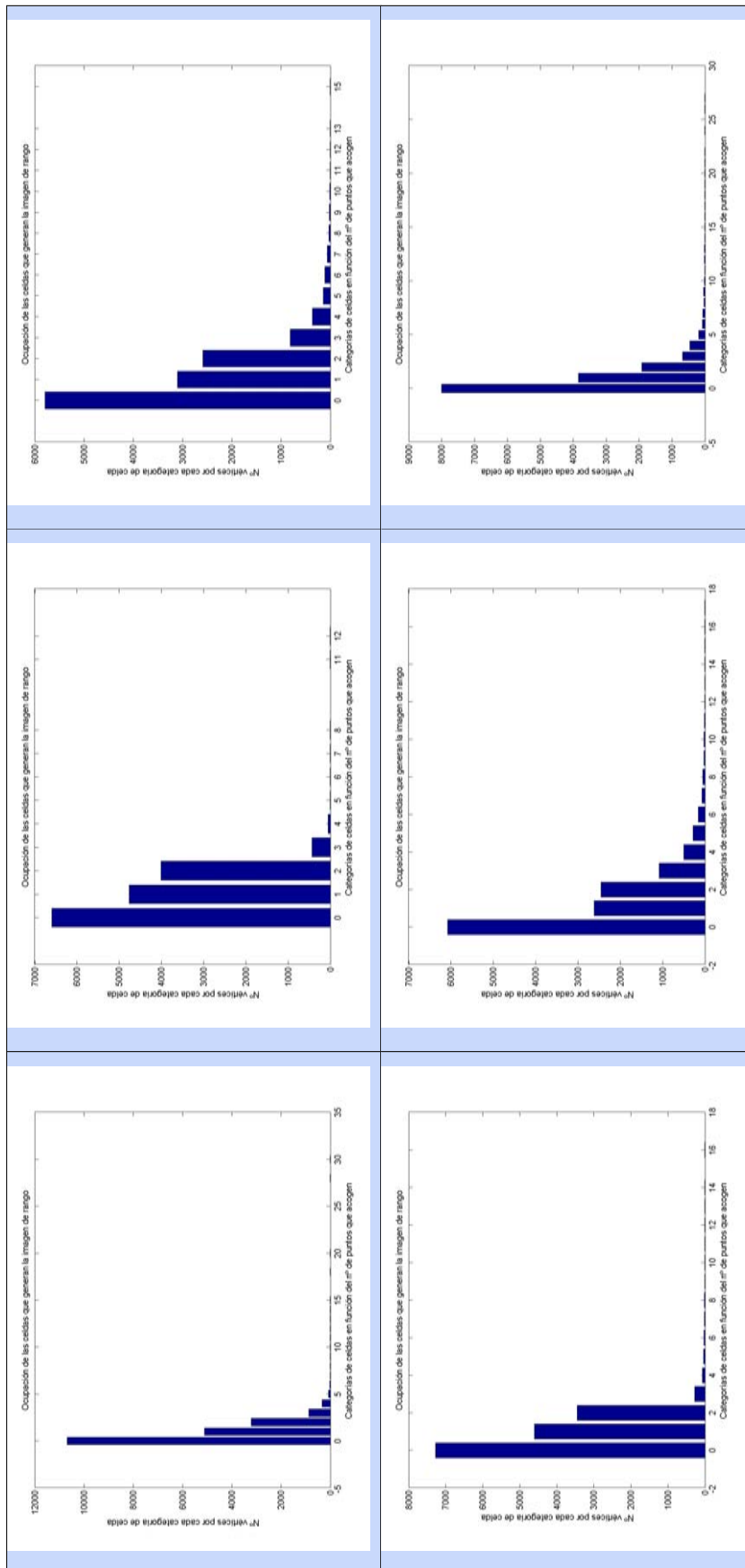


Tabla 7.1: (continuación)

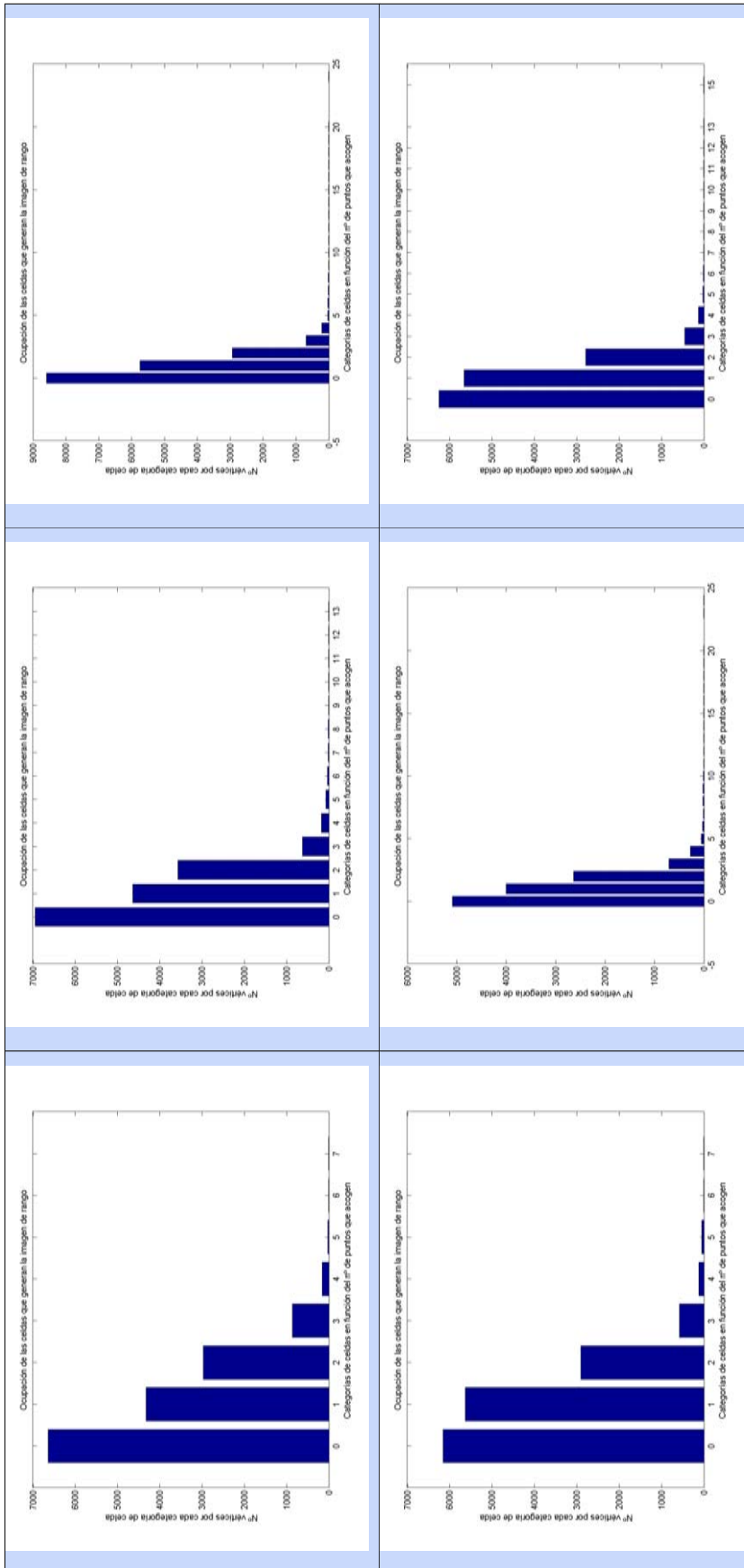
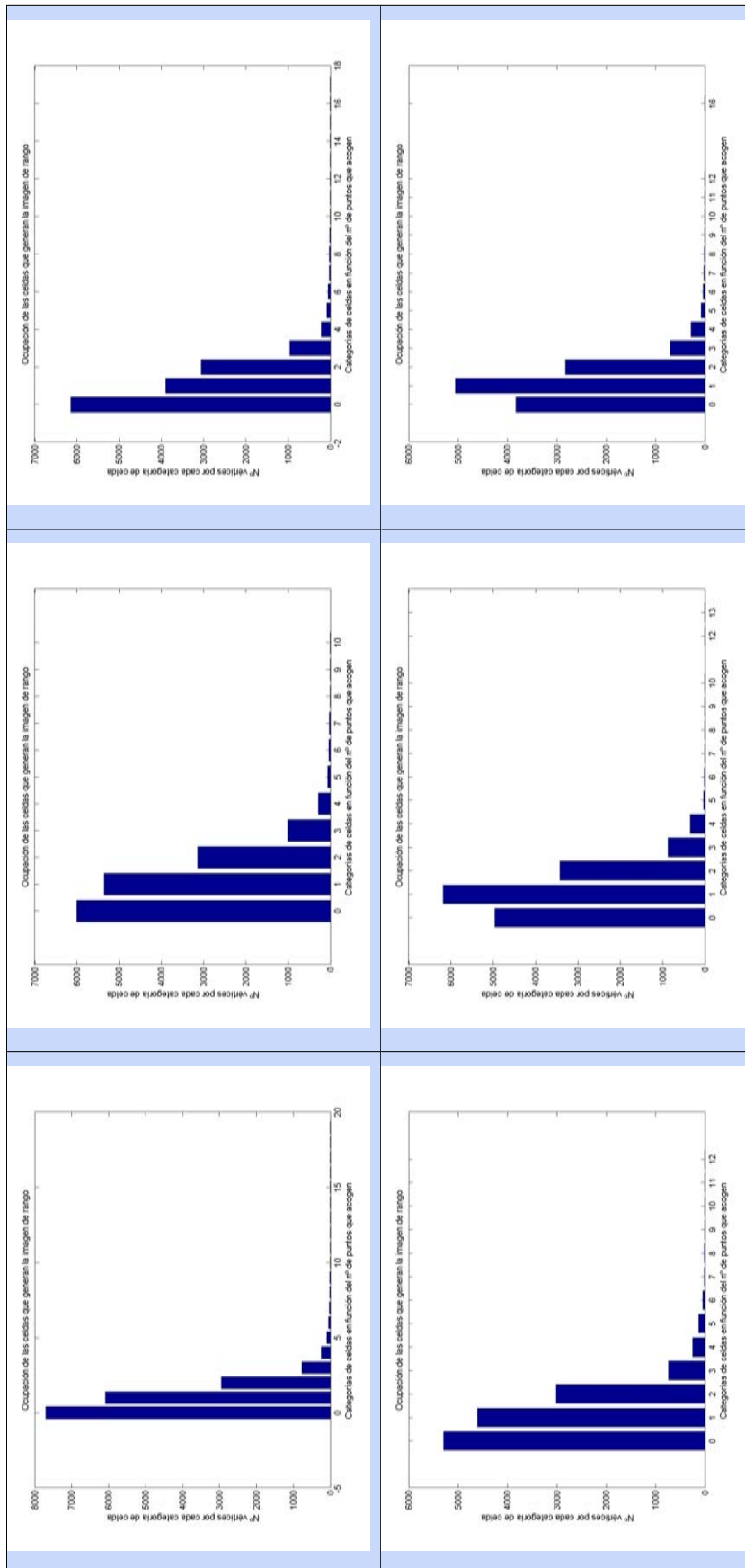


Tabla 7.1: (continuación)



7.7. Generación de la imagen de rango sin rasterización

Tabla 7.2: Generación de la imagen de rango sin rasterización

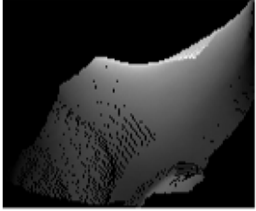
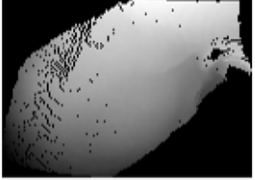
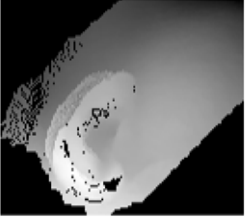
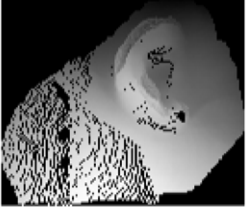
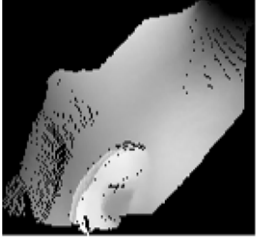

Generación de huecos artificiales	
	
	
	

Tabla 7.2: (continuación)

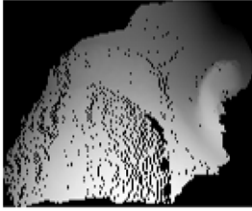
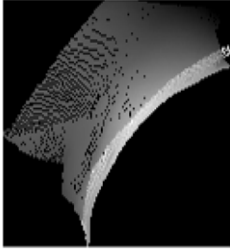

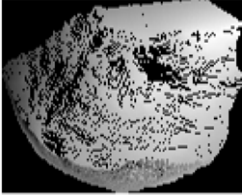
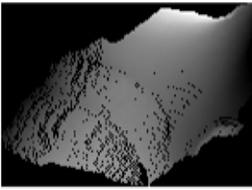
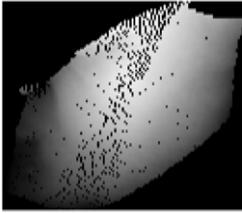
	
	
	

Tabla 7.2: (continuación)

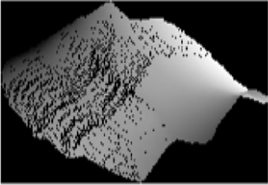
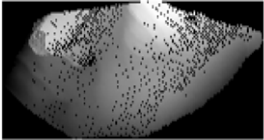
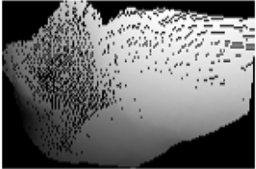
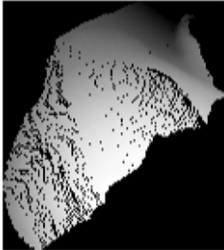
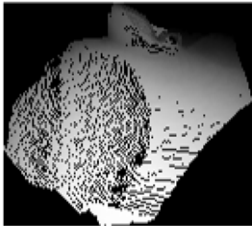
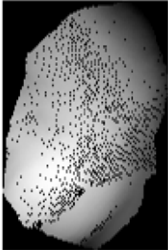
	
	
	

Tabla 7.2: (continuación)

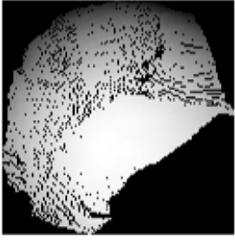

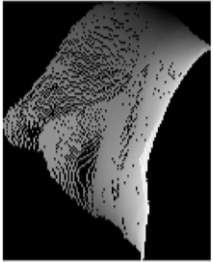
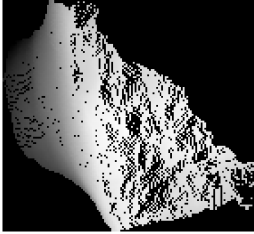
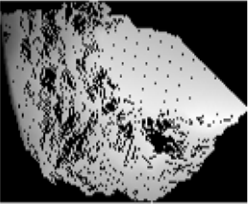

	
	
	

Tabla 7.2: (continuación)

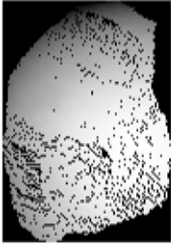
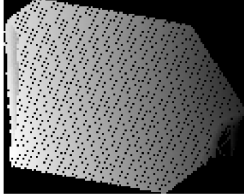
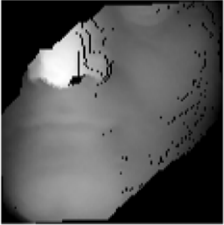
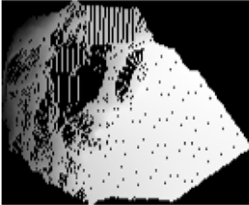
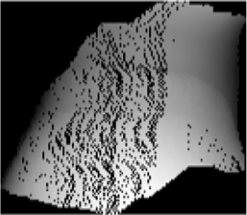
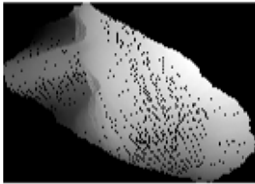
	
	
	

Tabla 7.2: (continuación)

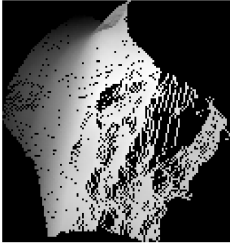
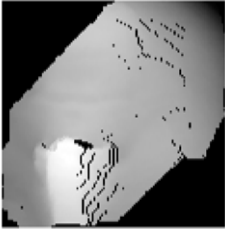
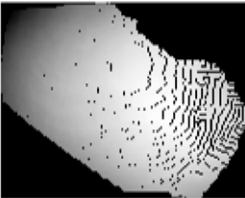
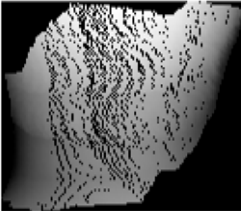
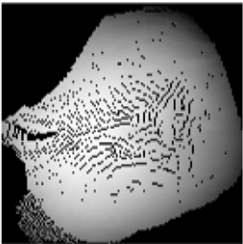
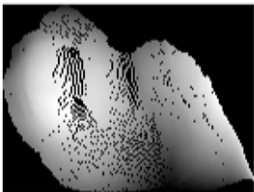
	
	
	

Tabla 7.2: (continuación)

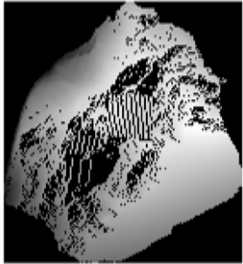
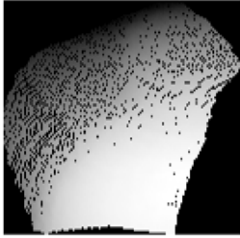
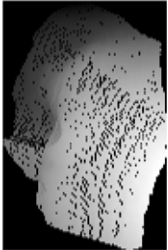
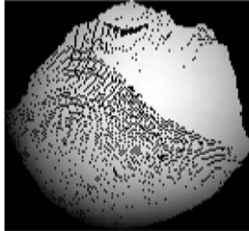
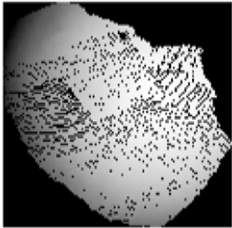
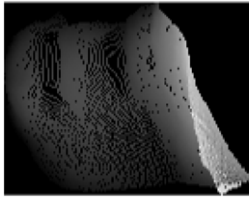
	
	
	

Tabla 7.2: (continuación)

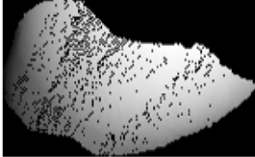
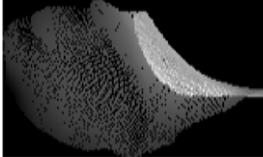
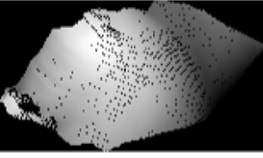
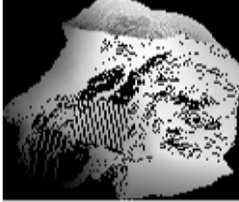
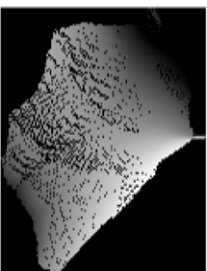
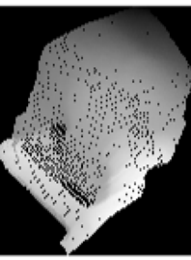
	
	
	

Tabla 7.2: (continuación)

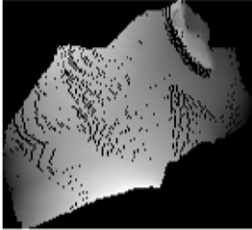
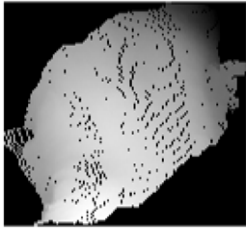
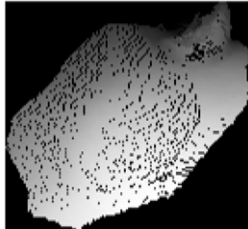

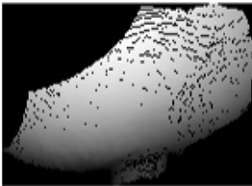
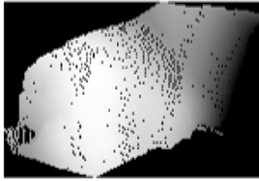
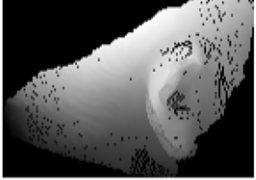
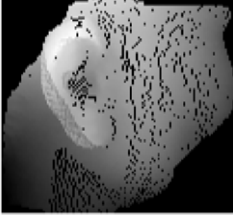
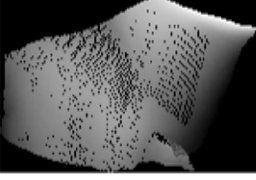
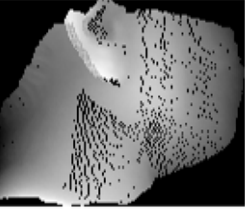
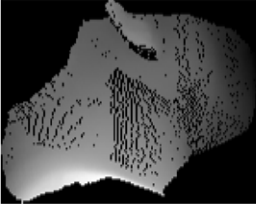
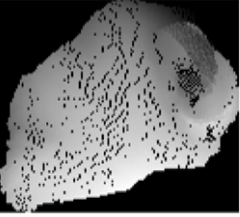
	
	
	

Tabla 7.2: (continuación)

7.8. Generación de la imagen de rango con rasterización

Tabla 7.3: Generación de la imagen de rango con rasterización


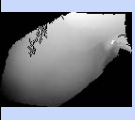

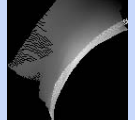
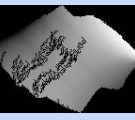
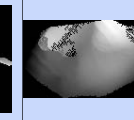
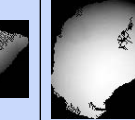
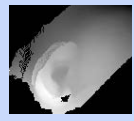
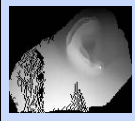
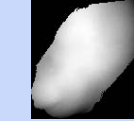
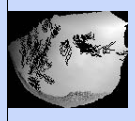
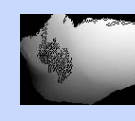
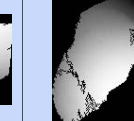
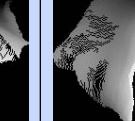



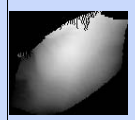
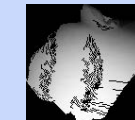
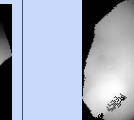
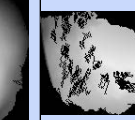
Generación de huecos artificiales							
							
							

Tabla 7.3: (continuación)



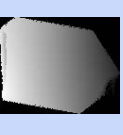
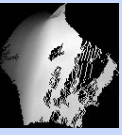

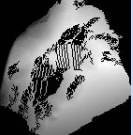


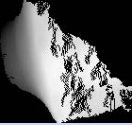
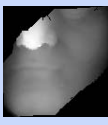
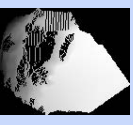

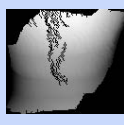
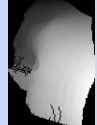
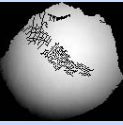
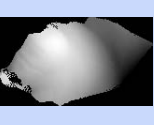
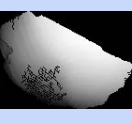
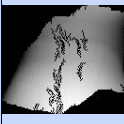

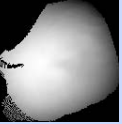


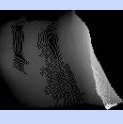

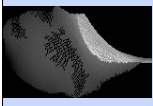

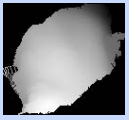

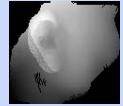

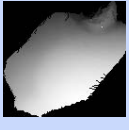








							
							
							

Tabla 7.3: (continuación)

7.9. Extensión de la porción de malla

Tabla 7.4: Extensión de la porción de malla

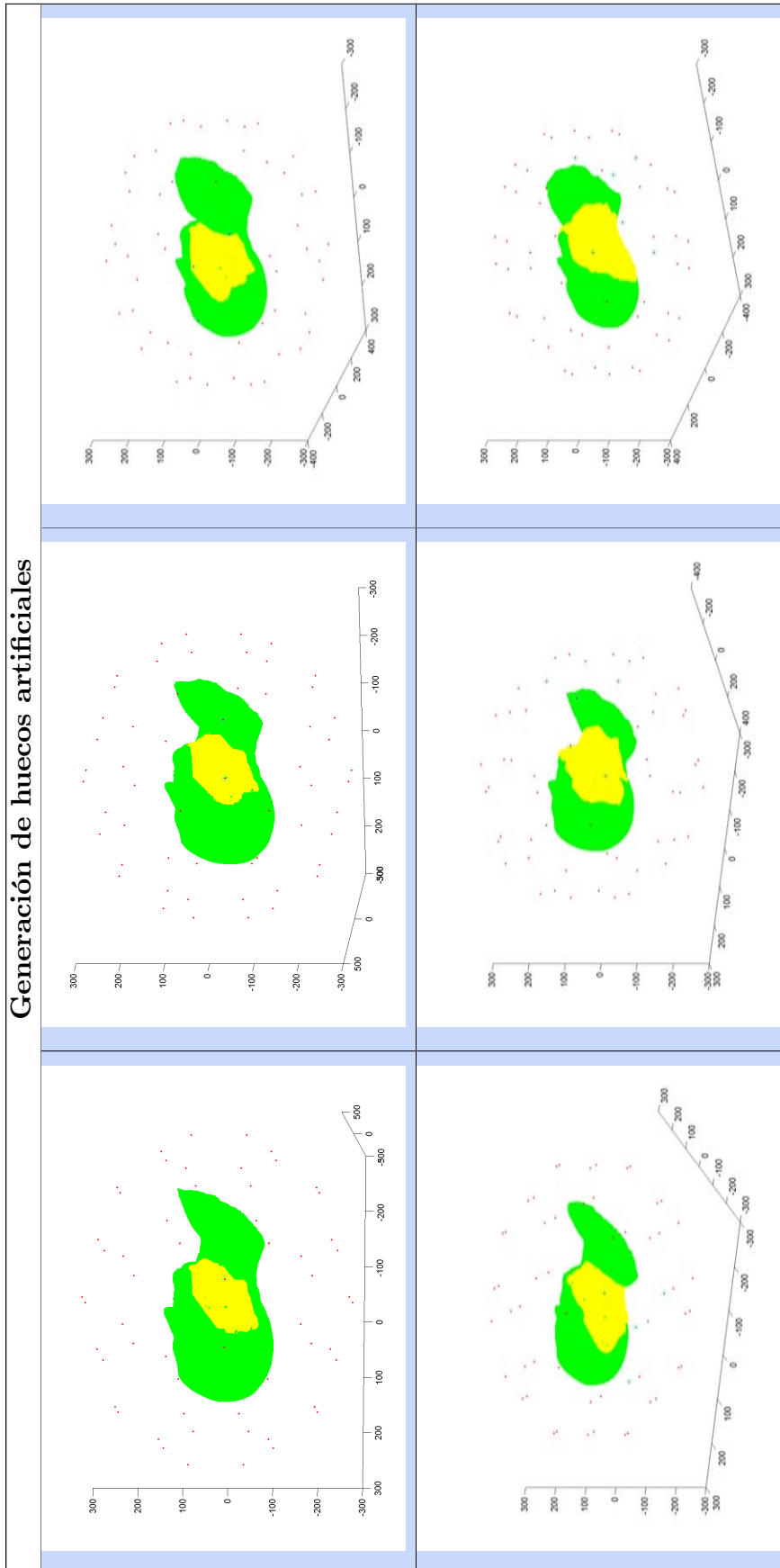


Tabla 7.4: (continuación)

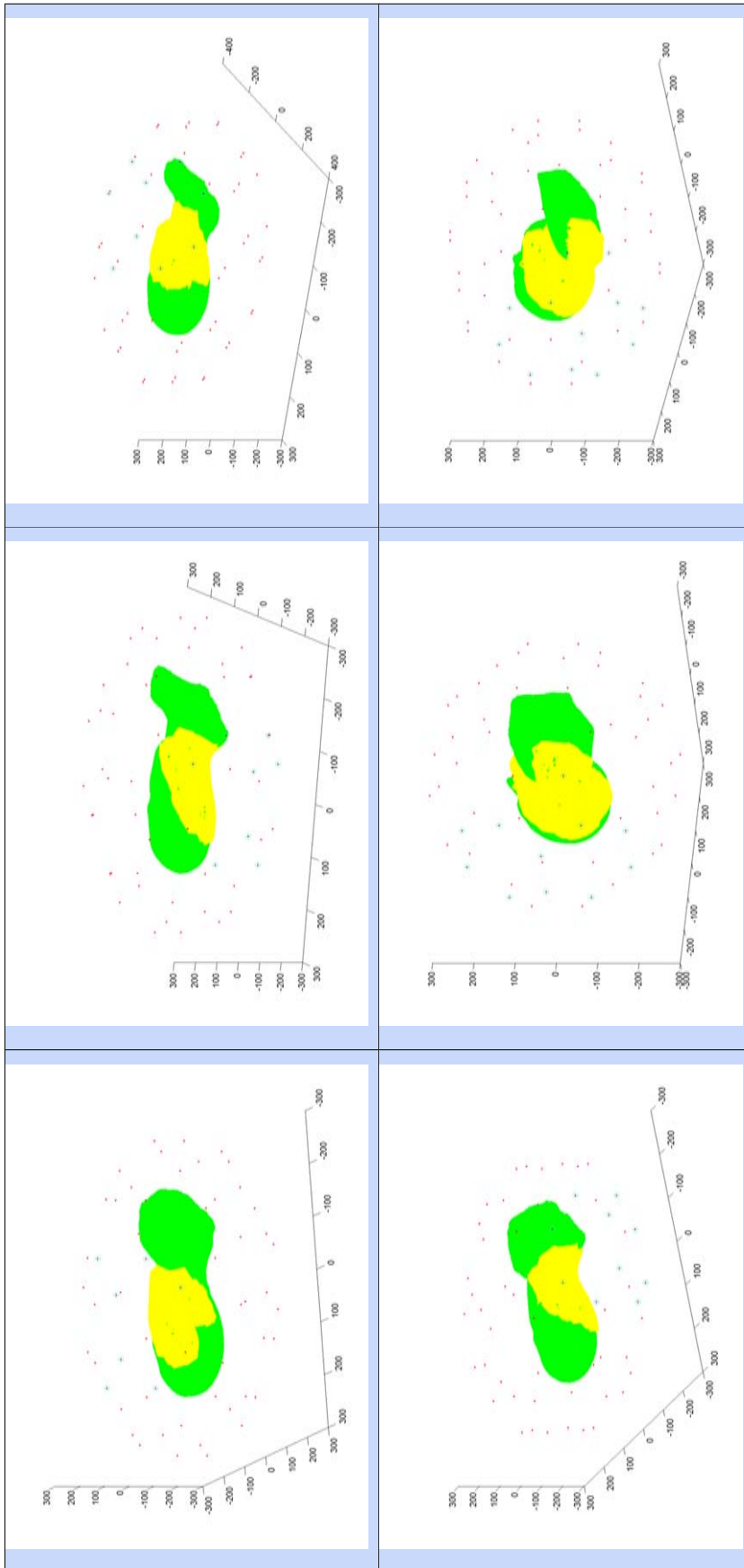


Tabla 7.4: (continuación)

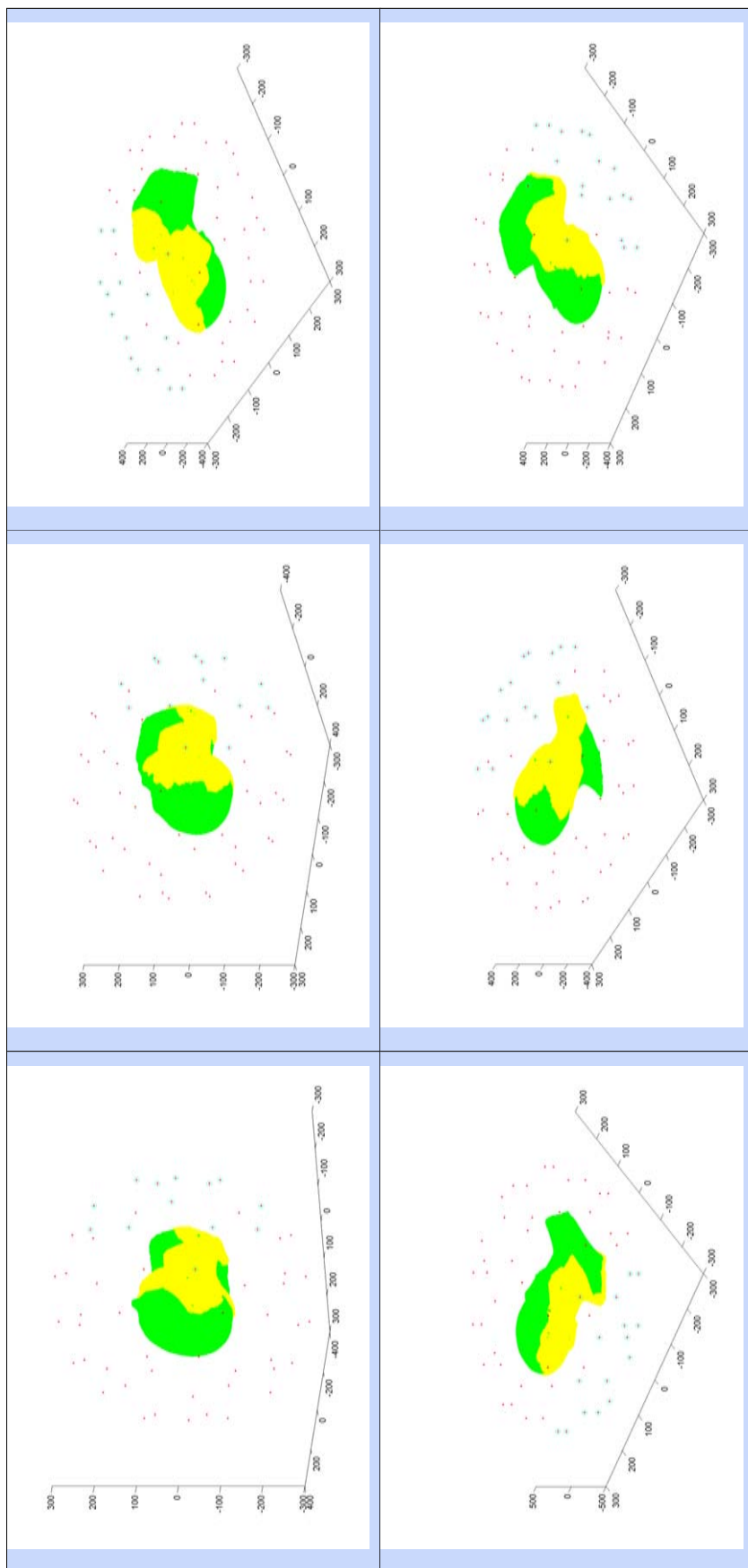


Tabla 7.4: (continuación)

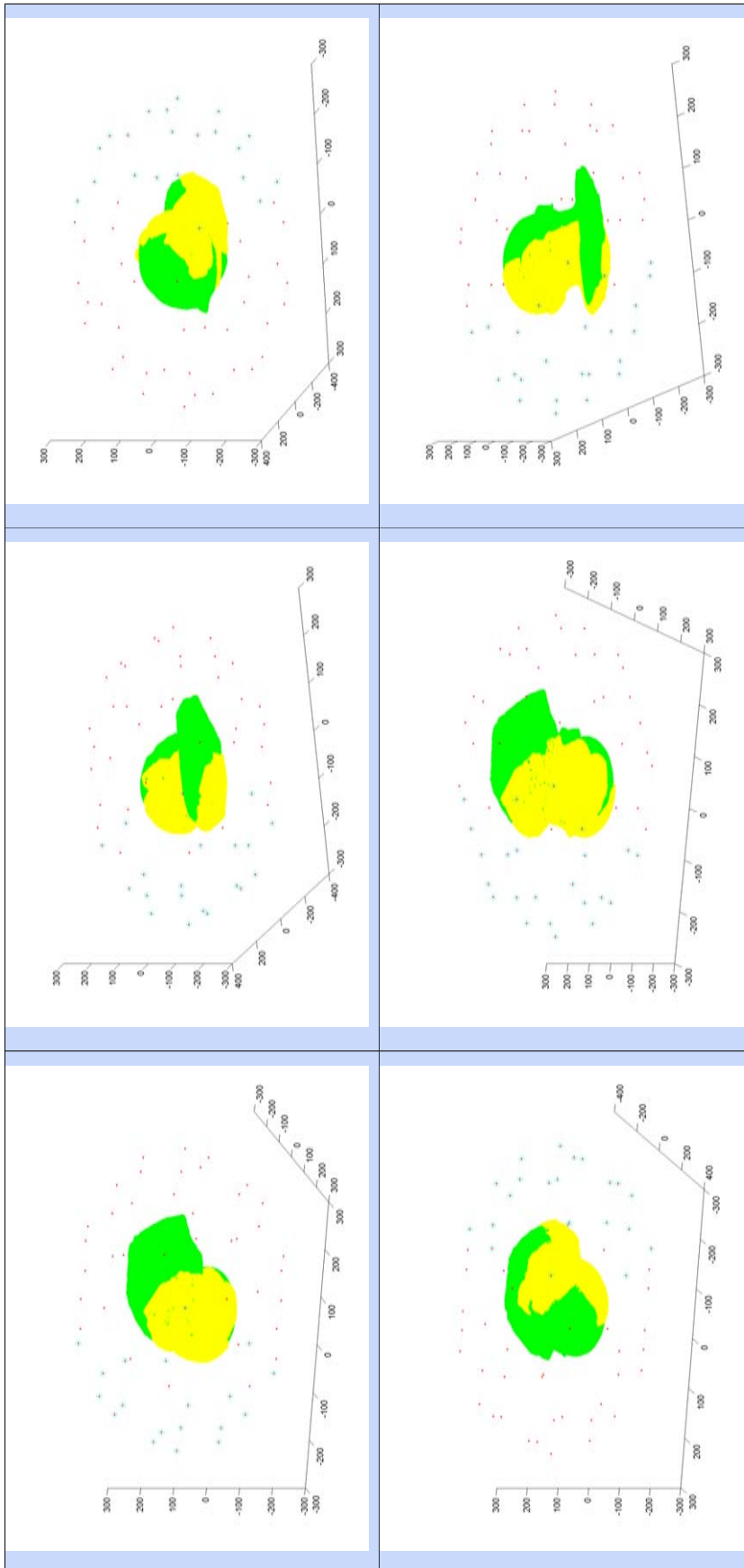


Tabla 7.4: (continuación)

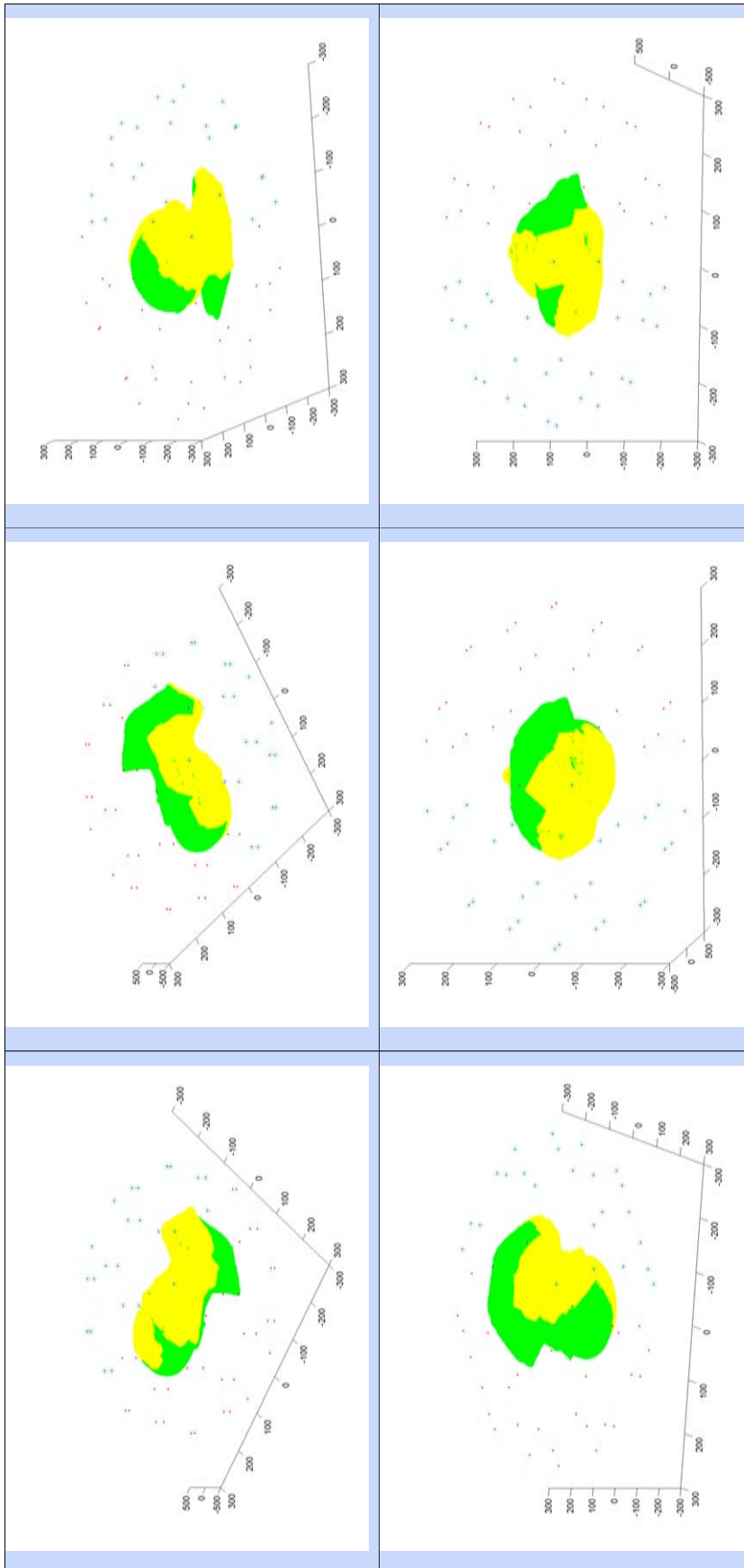


Tabla 7.4: (continuación)

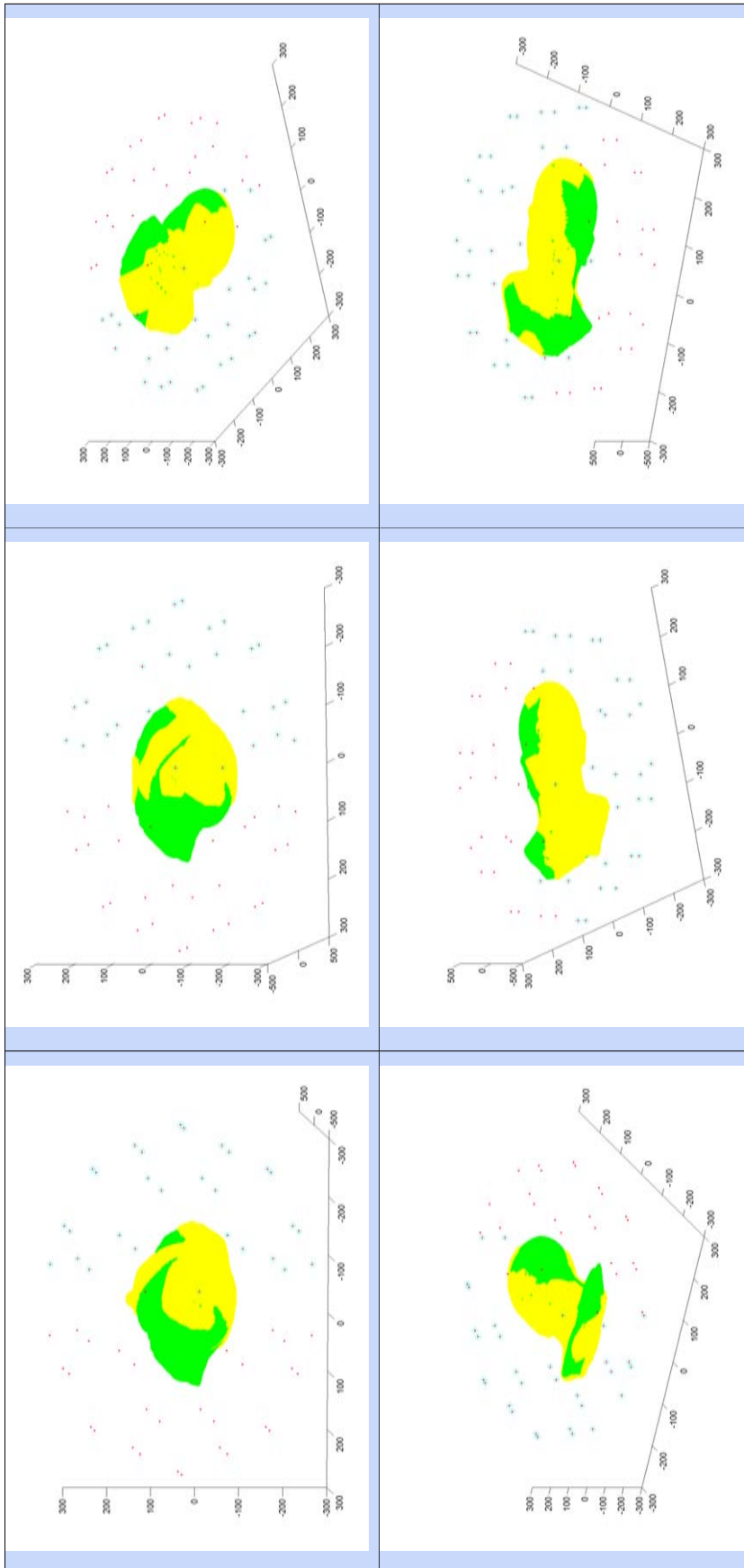


Tabla 7.4: (continuación)

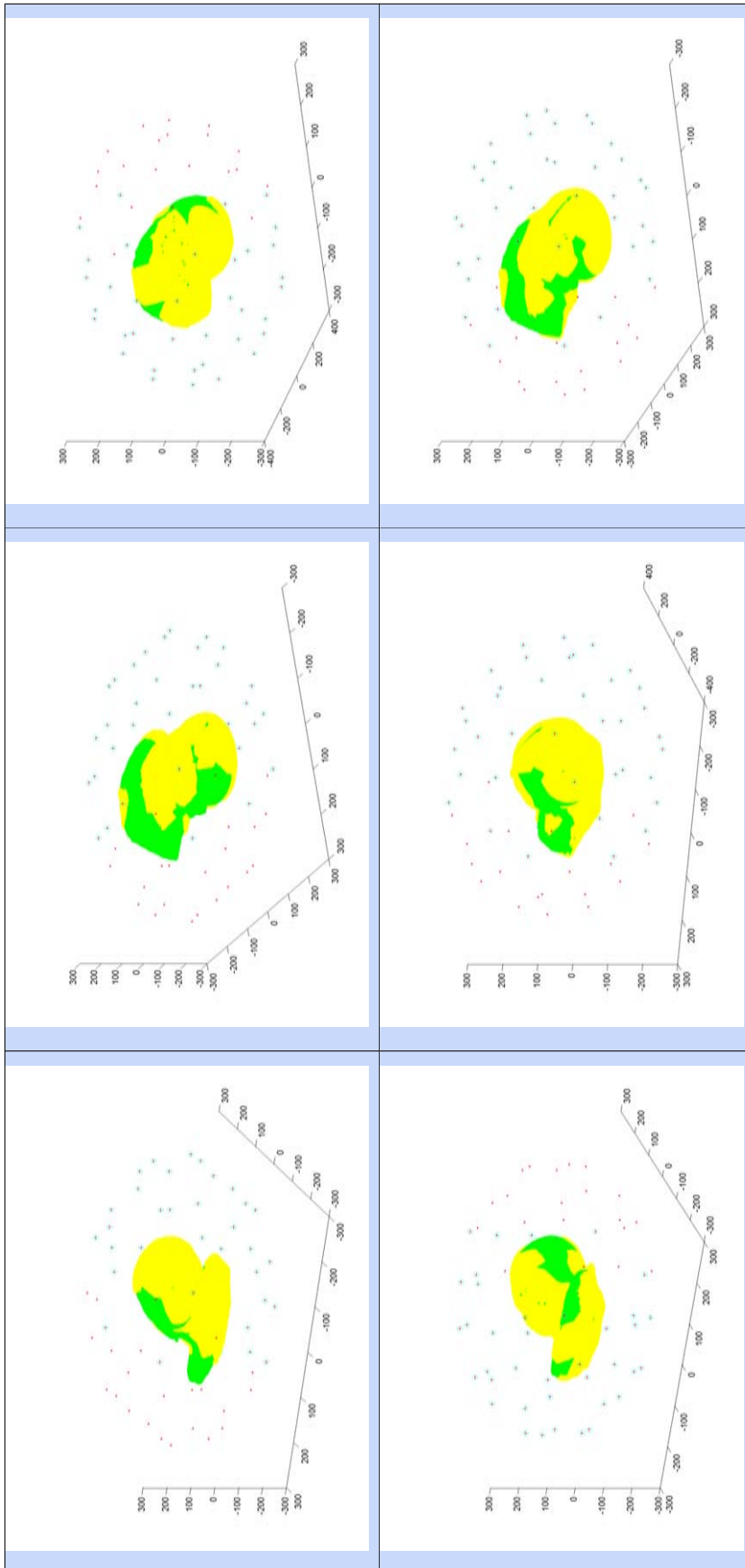


Tabla 7.4: (continuación)

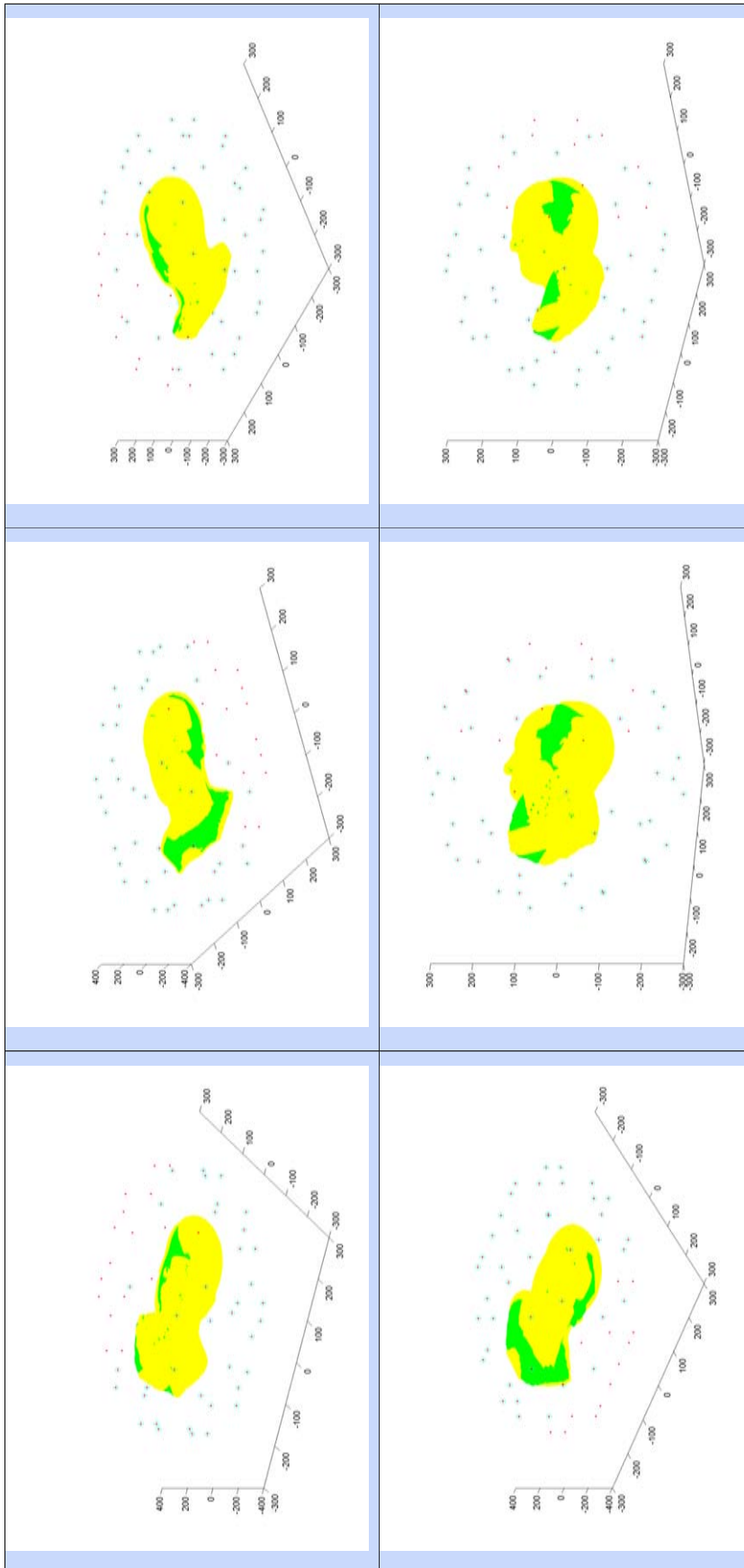


Tabla 7.4: (continuación)

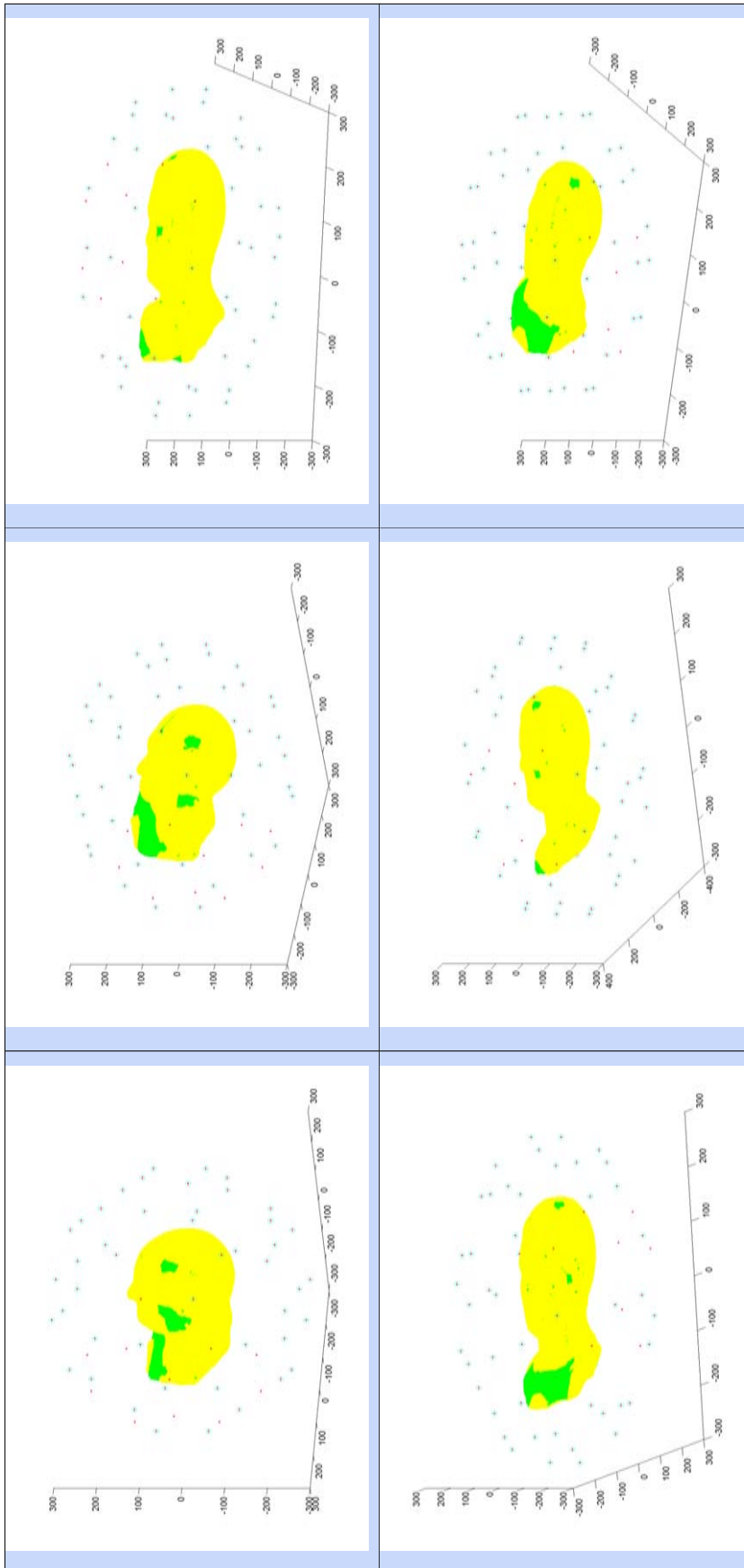
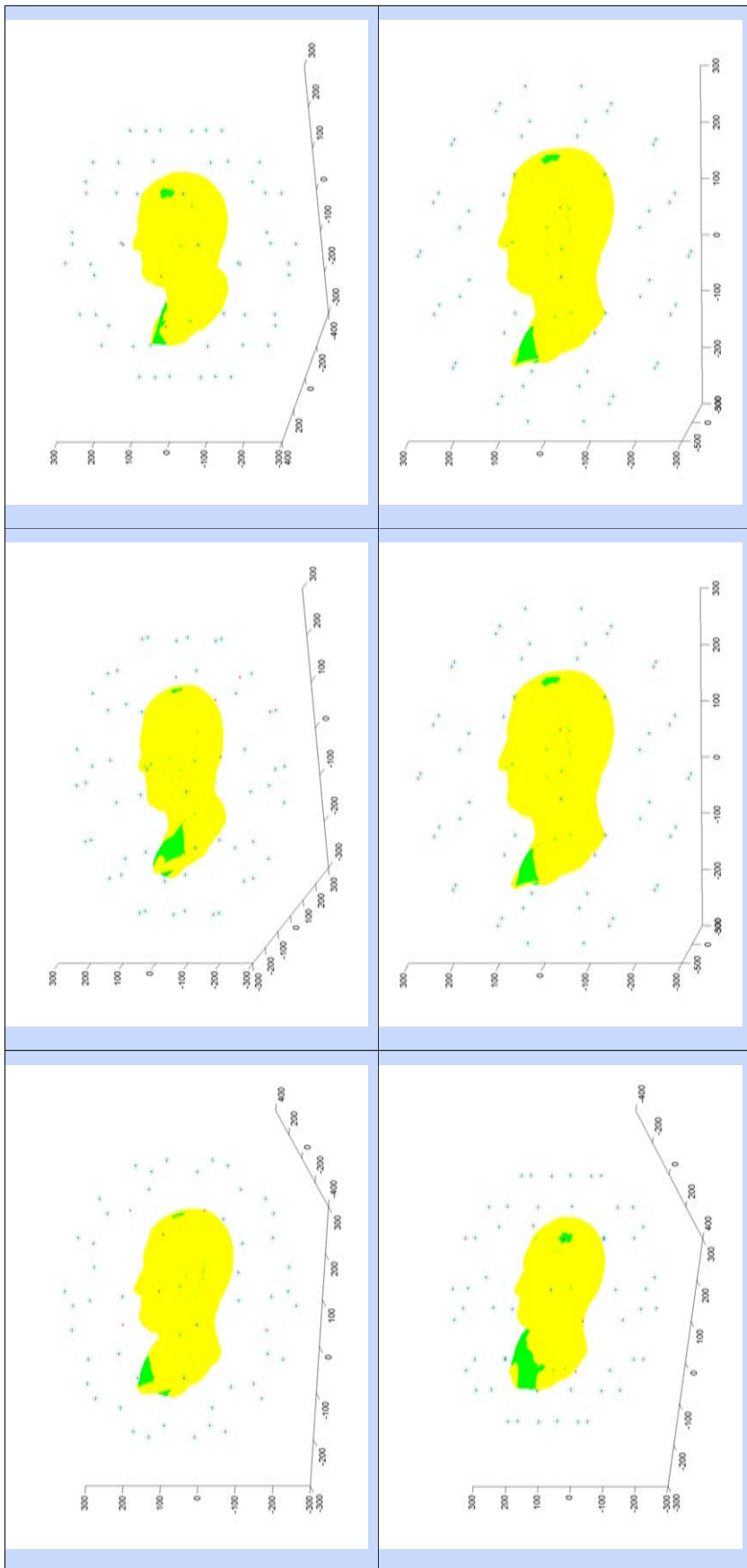


Tabla 7.4: (continuación)



7.10. Resultados experimentales para mallas completas

Tabla 7.5: Resultados experimentales para mallas completas

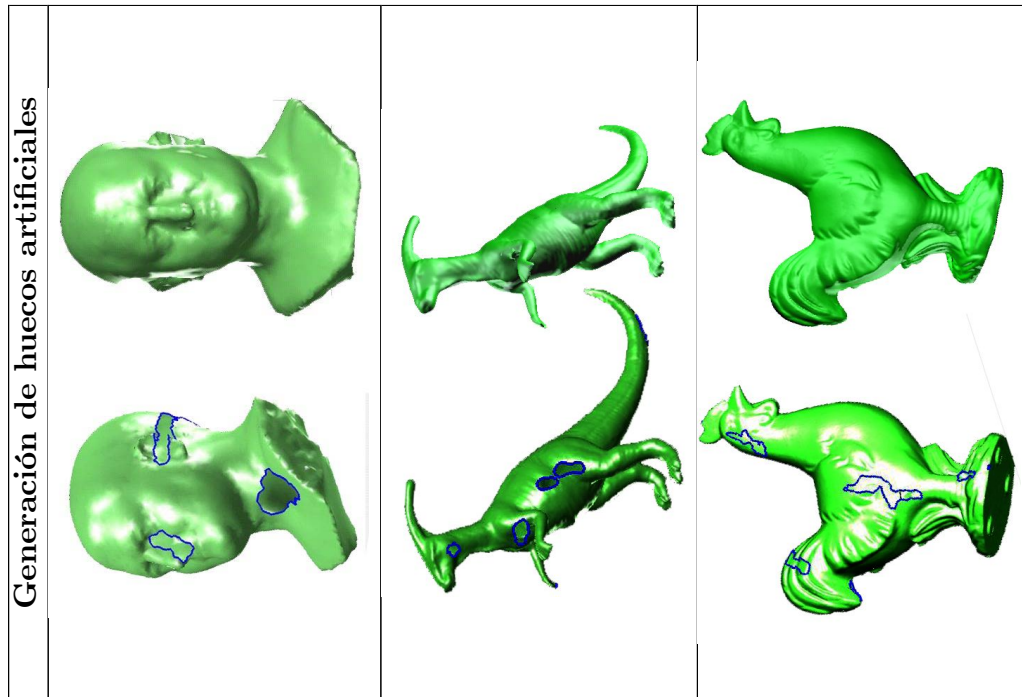
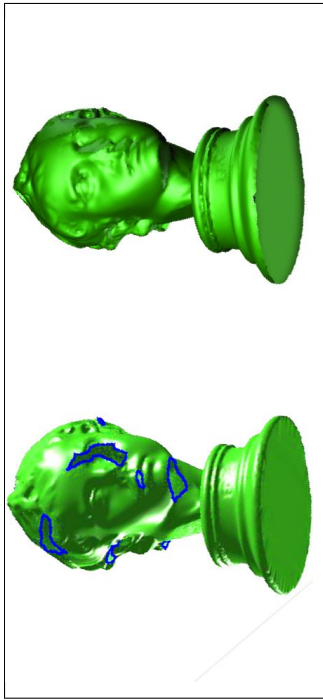


Tabla 7.5: (continuación)



7.11. Estudio de la curvatura antes y después del relleno

Tabla 7.6: Estudio de la curvatura antes y después del relleno

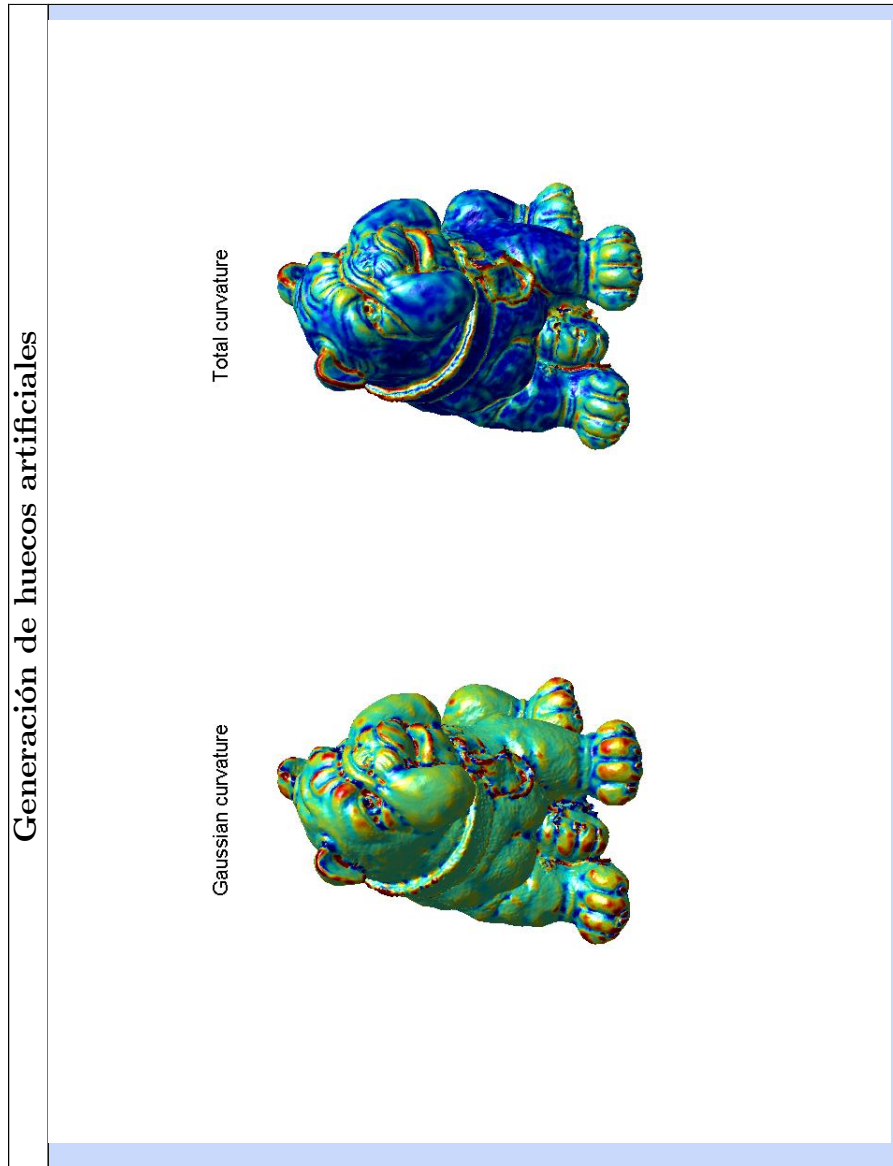


Tabla 7.6: (continuación)

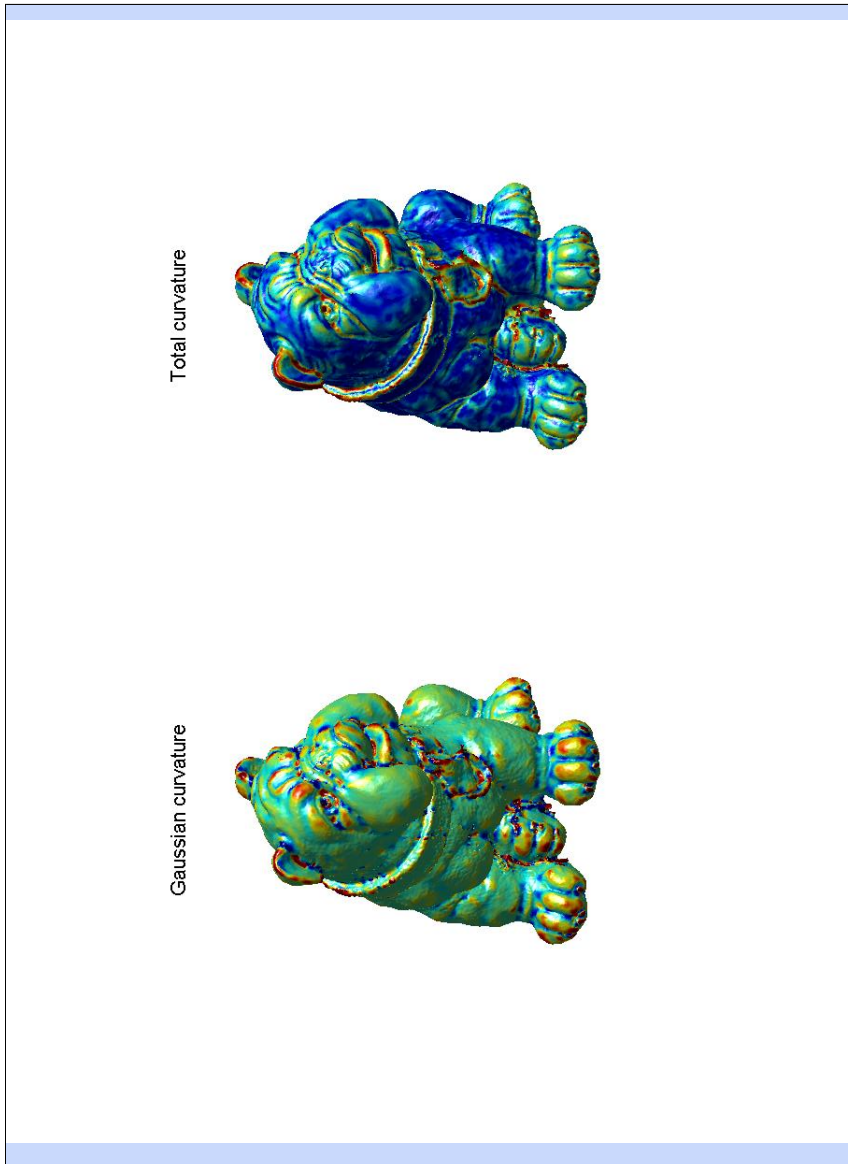


Tabla 7.6: (continuación)

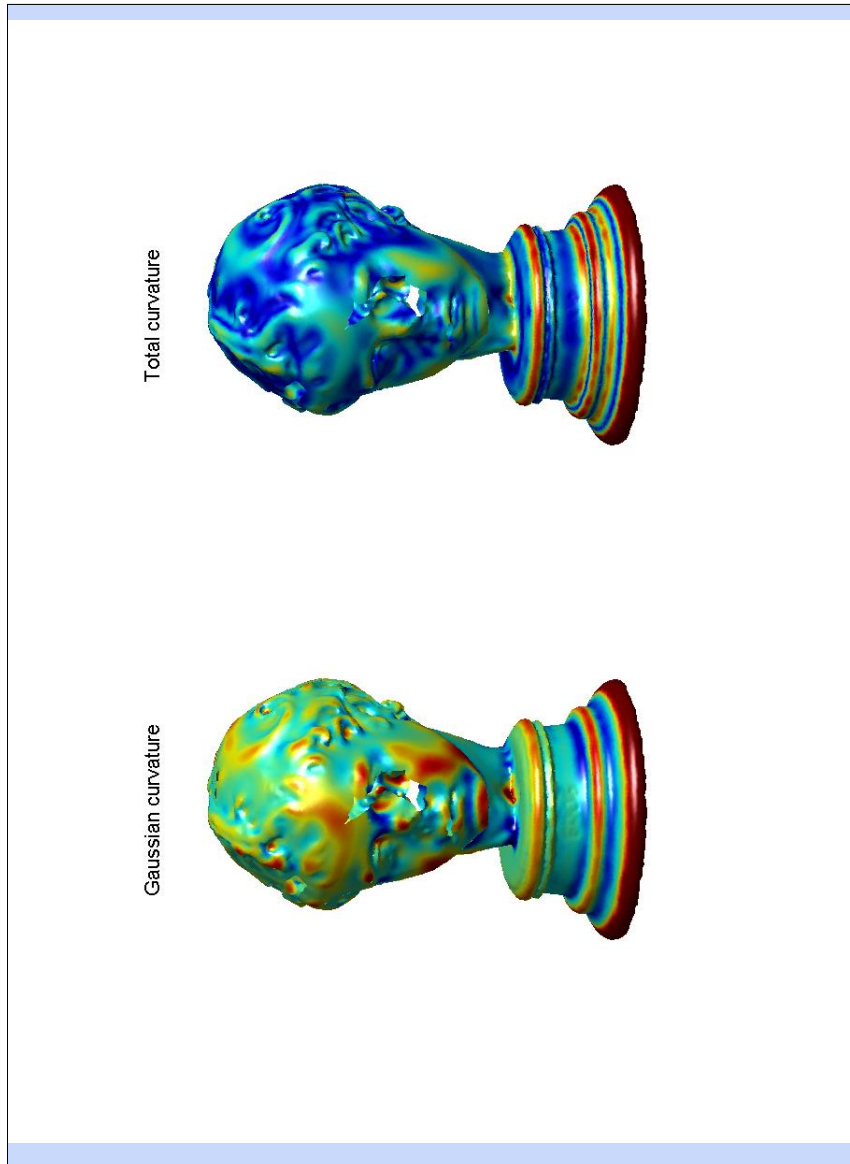


Tabla 7.6: (continuación)

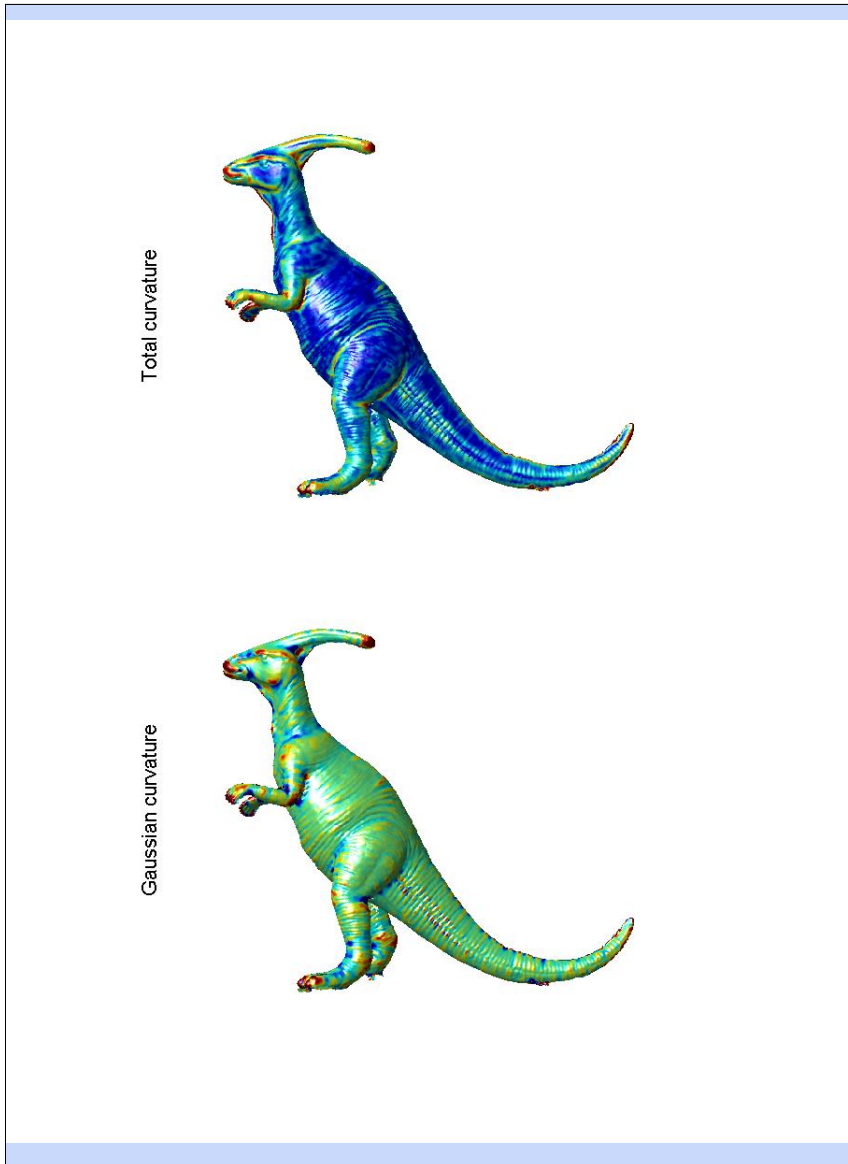
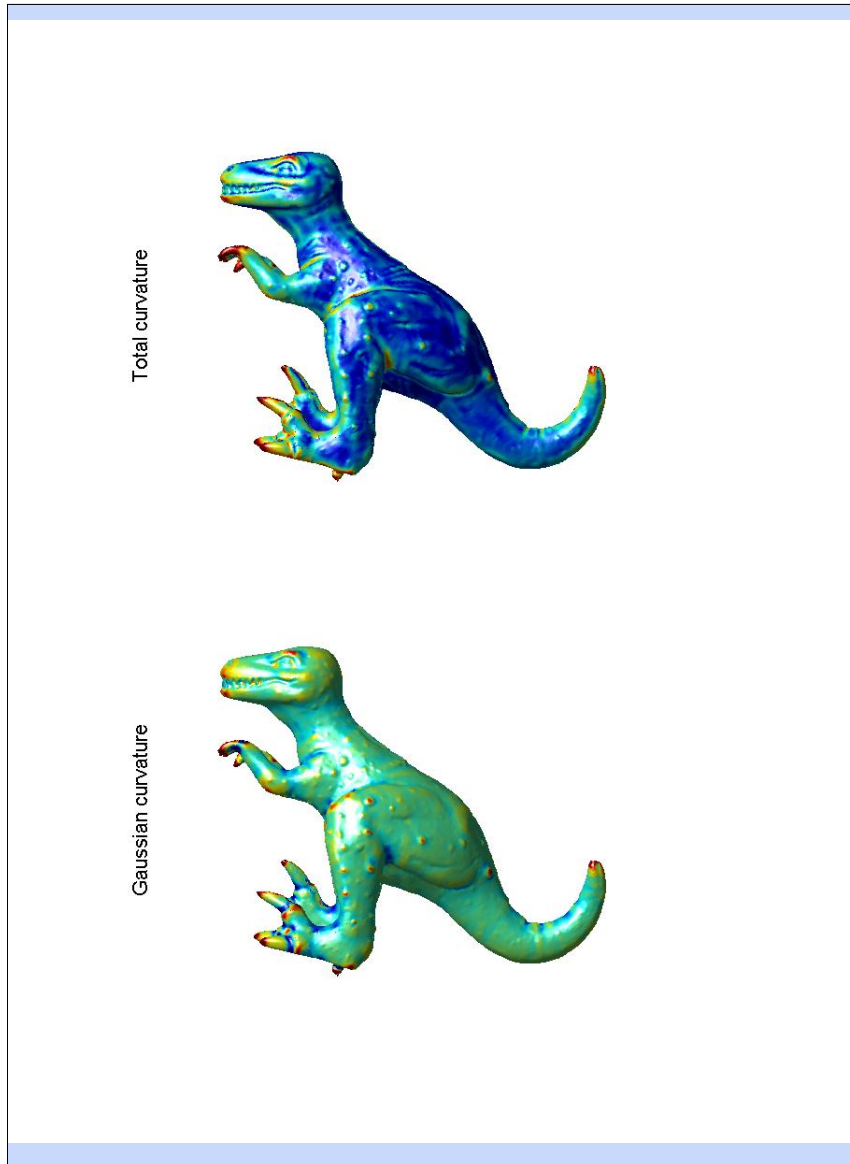


Tabla 7.6: (continuación)



Capítulo 8

Referencias

Bibliografía

- [Ame98] N. Amenta, M. Bern y M. Kamvyselis. A new Voronoi-based surface reconstruction algorithm. En: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98)*, pp. 415–421, Nueva York, EEUU, 1998.
- [Baj95] C.L. Bajaj, F. Bernardini y G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. En: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'95)*, pp. 109–118, Nueva York, EEUU, 1995.
- [Ber99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva y G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [Bes90] P.J. Besl. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes*, pp. 25–71, 1990.
- [Bes92] P.J. Besl y N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [Bli82] J.F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.
- [Blo97] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, EEUU, 1997.
- [Bot06] M. Botsch, M. Pauly, C. Rossl, S. Bischoff y L. Kobbelt. Geometric modeling based on triangle meshes. En: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'06)*, Nueva York, EEUU, 2006.
- [Bra06] J. Branch, F. Prieto y P. Boulanger. A hole-filling algorithm for triangular meshes using local radial basis function. En: *Proceedings of the 15th International Meshing Roundtable*, pp. 411–431, Alabama, EEUU, 2006.
- [Bre05] T.P. Breckon y R.B. Fisher. Non-parametric 3D surface completion. En: *Proceedings of the 5th International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pp. 573–580, Washington, EEUU, 2005.
- [Bru07] A. Brunton, S. Wuhler y C. Shu. Image-based model completion. En: *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling (3DIM'07)*, pp. 305–311, Washington, EEUU, 2007.

- [Bru09] A. Brunton, S. Wuhler, C. Shu, P. Bose y E.D. Demaine. Filling holes in triangular meshes by curve unfolding. En: *Proceedings of the 2009 IEEE International Conference on Shape Modeling and Applications (SMI'09)*, pp. 66–72, 2009.
- [Cam01] R.J. Campbell y P.J. Flynn. A survey of free-form object representation and recognition techniques. *Journal of Computer Vision and Image Understanding*, 81(2):166–210, 2001.
- [Can88] J.F. Canny. *The complexity of robot motion planning*. MIT Press, Cambridge, Inglaterra, 1988.
- [Can02] P. Cano y J.C. Torres. Representation of polyhedral objects using sp-octrees. *Journal of WSCG*, 10:95–101, 2002.
- [Car01] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum y T.R. Evans. Reconstruction and representation of 3D objects with radial basis functions. En: *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*, pp. 67–76, 2001.
- [Cas02] U. Castellani, S. Livatino y R.B. Fisher. Improving environment modelling by edge occlusion surface completion. En: *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*, 2002.
- [Cor07] N.D. Cornea, D. Silver y P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548, 2007.
- [Cur96] B.L. Curless y M. Levoy. A volumetric method for building complex models from range images. En: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96)*, pp. 303–312, Nueva York, EEUU, 1996.
- [Cur97] B.L. Curless. New methods for surface reconstruction from range images. Technical report, Stanford University, Stanford, EEUU, 1997.
- [Dav01] J. Davis, S.R. Marschner, M. Garr y M. Levoy. Filling holes in complex surfaces using volumetric diffusion. En: *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 428–438, 2001.
- [Del02] F. Dell'acqua y R.B. Fisher. Reconstruction of planar surfaces behind occlusions in range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:569–575, 2002.
- [Dey01] T.K. Dey, J. Giesen y J. Hudson. Delaunay based shape reconstruction from large data. En: *Proceedings of the IEEE 2001 Symposium on Parallel and Large-data Visualization and Graphics (PVG'01)*, pp. 19–27, Piscataway, EEUU, 2001.
- [Din01] H.Q. Dinh y G. Turk. Reconstructing surfaces using anisotropic basis functions. En: *Proceedings of the 2001 International Conference on Computer Vision (ICCV'01)*, pp. 606–613, 2001.
- [Ede94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

- [Efr99] A. Efros, A.A. Efros y T.K. Leung. Texture synthesis by non-parametric sampling. En: *Proceedings of the 2001 International Conference on Computer Vision (ICCV'01)*, pp. 1033–1038, 1999.
- [Guo06] T.-Q. Guo, J.-J. Li, J.-G. Weng y Y.-T. Zhuang. Filling holes in complex surfaces using oriented voxel diffusion. En: *Proceedings of the 5th International Conference on Machine Learning and Cybernetics*, 2006.
- [He06] X.J. He y Y.H. Chen. A haptics-guided hole-filling system based on triangular mesh. *Journal of Computer-Aided Design and Applications*, 3(6):711–718, 2006.
- [Igl02] A. Iglesias, A. Gálvez, J. Puig-Pey y F. Gutiérrez. Métodos computacionales para interrogación de superficies NURBS. Aplicaciones en la industria del automóvil. *Métodos Numéricos en Ingeniería y Ciencias Aplicadas*, 2:1055–1064, 2002.
- [Ju09] T. Ju. Fixing geometric errors on polygonal models: a survey. *Journal of Computer Science Technology*, 24(1):19–29, 2009.
- [Jun05] Y. Jun. A piecewise hole filling algorithm in reverse engineering. *Journal of Computer-Aided Design*, 37(2):263–270, 2005.
- [Ker04] D. Keren. Topologically faithful fitting of simple closed curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:118–123, 2004.
- [Kob98] L. Kobbelt, S. Campagna, J. Vorsatz y H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. En: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98)*, pp. 105–114, Nueva York, EEUU, 1998.
- [Kum07] A. Kumar, A.M. Shih, Y. Ito, D.H. Ross y B.K. Soni. A hole-filling algorithm using non-uniform rational B-splines. En: *Proceedings of the 16th International Meshing Roundtable*, pp. 169–182, 2007.
- [Lan81] P. Lancaster y K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.
- [Lev76] J. Levin. A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces. *Communication of the ACM*, 19(10):555–563, 1976.
- [Li08] G. Li, X. Ye y Sanyuan Zhang. An algorithm for filling complex holes in reverse engineering. *Engineering with Computers*, 24(2):119–125, 2008.
- [Li10] Z. Li, D.S. Meek y D.J. Walton. Polynomial blending in a mesh hole-filling application. *Journal of Computer-Aided Design*, 42(4):340–349, 2010.
- [Mar07] R. Marchand, J.Y. Lu, K. Kabin y R. Rankin. Unstructured meshes and finite elements in space plasma modelling: Principles and applications. *Advanced Methods for Space Simulations*, pp. 111–143, 2007.
- [Noc00] J. Nocedal y S.J. Wright. *Numerical Optimization*. Springer, 2000.
- [Noo03] F.S. Nooruddin y G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003.

- [Pan10] N. Pantuwong y M. Sugimoto. Skeleton-growing: a vector-field-based 3D curve-skeleton extraction algorithm. En: *Proceedings of the 37th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'10)*, Nueva York, EEUU, 2010.
- [Pav07] G. Pavlidis, A. Koutsoudis, F. Arnaoutoglou, V. Tsioukas y C. Chamzas. Methods for 3D digitization of cultural heritage. *Journal of Cultural Heritage*, 8(1):93–98, 2007.
- [Pod05] J. Podolak y S. Rusinkiewicz. Atomic volumes for mesh completion. En: *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP'05)*, Aire-la-Ville, Suiza, 2005.
- [Req80] A.A.G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys*, 12:437–464, 1980.
- [Rog01] D.F. Rogers. *An introduction to NURBS: with historical perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, EEUU, 2001.
- [Ros99] J.R. Rossignac y A.A.G. Requicha. Solid modeling, 1999.
- [Sag08] R. Sagawa y K. Ikeuchi. Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):686–699, 2008.
- [Sam88] H. Samet y R.E. Webber. Hierarchical data structures and algorithms for computer graphics. Part I. *IEEE Computer Graphics and Applications*, 8:48–68, 1988.
- [Sev09] R. Sevilla. *NURBS-Enhanced Finite Element Method*. Tesis Doctoral, Universidad Politécnica de Catalunya, 2009.
- [Sha04] A. Sharf, M. Alexa y D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics*, 23(3):878–887, 2004.
- [Sor04] O. Sorkine y D. Cohen-Or. Least-squares meshes. En: *Proceedings of the 2004 International Conference on Shape Modeling Applications (SMI'04)*, pp. 191–199, 2004.
- [Stu01] F. Stulp y R. Fisher. Reconstruction of surfaces behind occlusions in range images. En: *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM'01)*, pp. 232–239, 2001.
- [Tau96] G. Taubin y R. Ronfard. Implicit simplicial models for adaptive curve reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:321–325, 1996.
- [Tur01] G. Turk, H.Q. Dinh, J.F. O'Brien y G. Yngve. Implicit surfaces that interpolate. En: *Proceedings of the 2001 International Conference on Shape Modeling and Applications (SMI'01)*, pp. 62–71, 2001.
- [Vas06] L. Vása. Methods for dynamic mesh size reduction. Tesis Doctoral, República Checa, 2006.

- [Ver75] K.J. Versprille. *Computer-aided design applications of the rational B-spline approximation form*. Tesis Doctoral, Syracuse, NY, EEUU, 1975.
- [Wan07] J. Wang y M.M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing*, 25(1):103–113, 2007.
- [Wu08] X.J. Wu, M.Y. Wang y B. Han. An automatic hole-filling algorithm for polygon meshes. *Journal of Computer-Aided Design and Applications*, 5(6):889–899, 2008.
- [Zha07] W. Zhao, S. Gao y H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007.