

Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics
Màster en Computació

Tesis de Máster

Inspección Interactiva de Estructuras Anatómicas Tubulares

Estudiante: José Díaz Iriberry
Director: Isabel Navazo Álvaro
Ponente: -

Fecha: 18 de julio de 2007

Contenido

1. Introducción	2
1.1 Visualización de estructuras anatómicas	2
1.2 Objetivos y estructura de la tesis	4
2. Etapas en la visualización médica	5
2.1 Introducción	5
2.2 Captación de datos médicos	6
2.3 Modelo de volumen	7
2.4 Segmentación de los datos	9
2.5 Clasificación de los datos	11
2.6 Visualización de los datos	12
2.6.1 Visualización de isosuperficies	13
2.6.2 Visualización Directa del Volumen	14
2.7 Etapas de una endoscopia virtual	19
2.8 Resumen	21
3. Estado del arte en endoscopia virtual	22
3.1 Introducción	22
3.2 Sistemas no comerciales	22
3.3 Sistemas comerciales	25
3.4 Conclusiones	27
4. Visualización de endoscopias virtuales con texturas 3D	30
4.1 Introducción	30
4.2 Proceso de visualización con texturas 3D	30
4.3 Análisis de las soluciones planteadas	34
4.4 Técnicas de optimización para la inspección interactiva	36
4.4.1 Adaptative sampling	36
4.4.2 Adaptación del viewport	39
4.4.3 Detección de colisiones	41
4.5 Discusión y evaluación de resultados	42
5. Conclusiones y trabajo futuro	53
Bibliografía	55
Anexo A	57
A.1 Características del sistema de endoscopia virtual	57
A.2 Shaders utilizados en el sistema	59

Capítulo 1

Introducción

endoscopia.

(De *endo-* y *-scopia*).

1. f. *Med.* Técnica de exploración visual de una cavidad o conducto del organismo.
2. f. *Med.* Esta exploración.

Diccionario de la Lengua Española. Real Academia Española. Vigésima segunda edición.

1.1 Visualización de estructuras anatómicas

Desde que en 1806, el alemán Philip Bozzini presentara el primer instrumento para visualizar los órganos internos del cuerpo humano hasta nuestros días, la endoscopia se ha convertido en un sistema de diagnóstico médico mínimamente invasivo utilizado, no sólo para mostrar imágenes del interior del organismo, sino también para realizar pequeñas operaciones quirúrgicas o extraer muestras de tejido.

La técnica consiste en la introducción de una sonda equipada con una pequeña cámara, ya sea por los orificios naturales del cuerpo humano o bien realizando una pequeña incisión. Para iluminar el interior, se transmite la luz generada por una fuente externa mediante un cable de fibra óptica y, a través de una lente, el especialista puede inspeccionar detenidamente la zona de interés.

Los sistemas de mínima invasión están adquiriendo una gran importancia en la medicina de los últimos años, debido a que los efectos producidos sobre el paciente son mucho menos perjudiciales que la cirugía abierta o convencional. A pesar de ello, el hecho de que haya que sedar o anestesiarse al paciente antes de la inspección, el riesgo de perforaciones, infecciones o hemorragias en la zona a tratar y la flexibilidad y navegación limitadas que permiten, son algunos de los inconvenientes de estos procesos.

Recientemente, gracias a los sistemas de captación de imágenes médicas existentes y a los progresos realizados en los gráficos por computador, se están desarrollando sistemas para realizar *endoscopias virtuales*. Así, a partir de imágenes obtenidas mediante tomografías, resonancias magnéticas u otras técnicas de adquisición de datos médicos, se puede inspeccionar un modelo virtual del interior del paciente, sin estar sujeto a las limitaciones de los procedimientos ópticos.

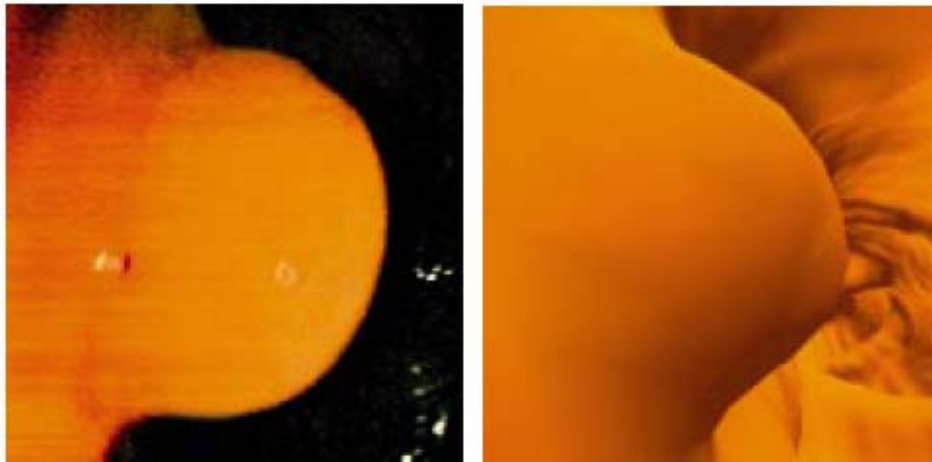
Para realizar un sistema de endoscopias virtuales se han de seguir tres etapas básicas: la adquisición y extracción de los datos, la visualización de las estructuras internas y la navegación por el interior de los órganos.

Actualmente, los sistemas de endoscopia virtual se están utilizando con diferentes propósitos:

- *Diagnóstico*: para la detección de anomalías en los órganos del paciente (por ejemplo la colonoscopia para detectar pólipos).
- *Educación*: para el entrenamiento de los cirujanos y practicantes.
- *Planificación pre-operatoria*: para examinar la zona a operar antes de realizar la intervención, planificar trayectorias de incisión, detectar las regiones a tratar, etc.
- *Apoyo durante la operación*: puede ser utilizada durante la intervención quirúrgica para obtener información complementaria.
- *Seguimiento post-operatorio*: para visualizar la evolución de un paciente después de una intervención.

La figura 1.1 muestra la misma vista del interior obtenida con una endoscopia convencional y con una virtual. Obsérvese que la imagen de la endoscopia virtual permite ver zonas que no son visibles con la endoscopia convencional.

Por otra parte, una de las ventajas que presentan los métodos virtuales es que permiten explorar zonas que las endoscopias convencionales no pueden mostrar, ya sea por el tamaño del elemento a observar o bien porque resulta difícil el acceso con la cámara (por ejemplo, se utilizan para realizar inspecciones de los vasos sanguíneos del cerebro).



*Figura 1.1: Vista obtenida con una endoscopia óptica (izq.) y con una virtual (der.).
Imagen extraída de [1].*

A pesar de las ventajas que presentan, las técnicas virtuales no pueden reemplazar totalmente a las técnicas ópticas en la práctica clínica. El hecho de estar condicionadas por la calidad y la resolución de las imágenes médicas, no poder observar ni los colores reales ni las texturas de los tejidos, no poder intervenir físicamente sobre las zonas visualizadas y la relación calidad – interactividad en la inspección de los datos, son algunas de las limitaciones que impiden su práctica clínica habitual.

1.2 Objetivos y estructura de la tesis

El objetivo que se persigue con esta tesis es analizar y evaluar los diferentes sistemas de visualización de datos volumétricos utilizados para realizar inspecciones virtuales de estructuras tubulares, así como proponer algunas soluciones a las limitaciones de estos métodos.

En el capítulo 2 se presentan las etapas principales en la visualización de datos de volumen: los sistemas de captación, el modelo de *voxels*, la segmentación o clasificación de los datos y las técnicas de visualización más utilizadas. Se concluye el capítulo introduciendo las etapas específicas de las que consta un sistema de endoscopias virtuales.

En el capítulo 3 se describen las características principales de los sistemas de endoscopia virtual, tanto los desarrollados por investigadores de la comunidad científica (presentados en artículos) como los disponibles en el ámbito comercial. También se analiza la utilización de cada una de las técnicas expuestas en el capítulo 2 y sus limitaciones.

En el capítulo 4 se exponen los aspectos a tener en cuenta cuando se utilizan texturas 3D como técnica de visualización en endoscopias virtuales. Se presentan los factores que pueden reducir la calidad de los resultados, se analizan las soluciones planteadas, se describen algunas propuestas propias para mejorar calidad y rendimiento, y se evalúan los resultados.

La memoria acaba con el capítulo 5 donde se presentan las conclusiones y el trabajo futuro. Se incluye un anexo donde se describe el sistema desarrollado para evaluar las técnicas analizadas en esta tesis y el código fuente de los *shaders* utilizados.

Capítulo 2

Etapas en la visualización médica

2.1 Introducción

Todo proceso destinado a la visualización de datos anatómicos comparte el conjunto de etapas mostradas en la figura 2.1. A partir de las imágenes obtenidas con la captación, se genera un modelo de volumen al que se le aplica un proceso de segmentación o de clasificación para identificar las diferentes estructuras anatómicas. A continuación se procede a la visualización, ya sea mediante mallas poligonales o bien de forma directa con *ray-casting* o texturas 3D. En este capítulo se pretende describir cada etapa, su objetivo y las diferentes técnicas con las que se puede llevar a cabo.

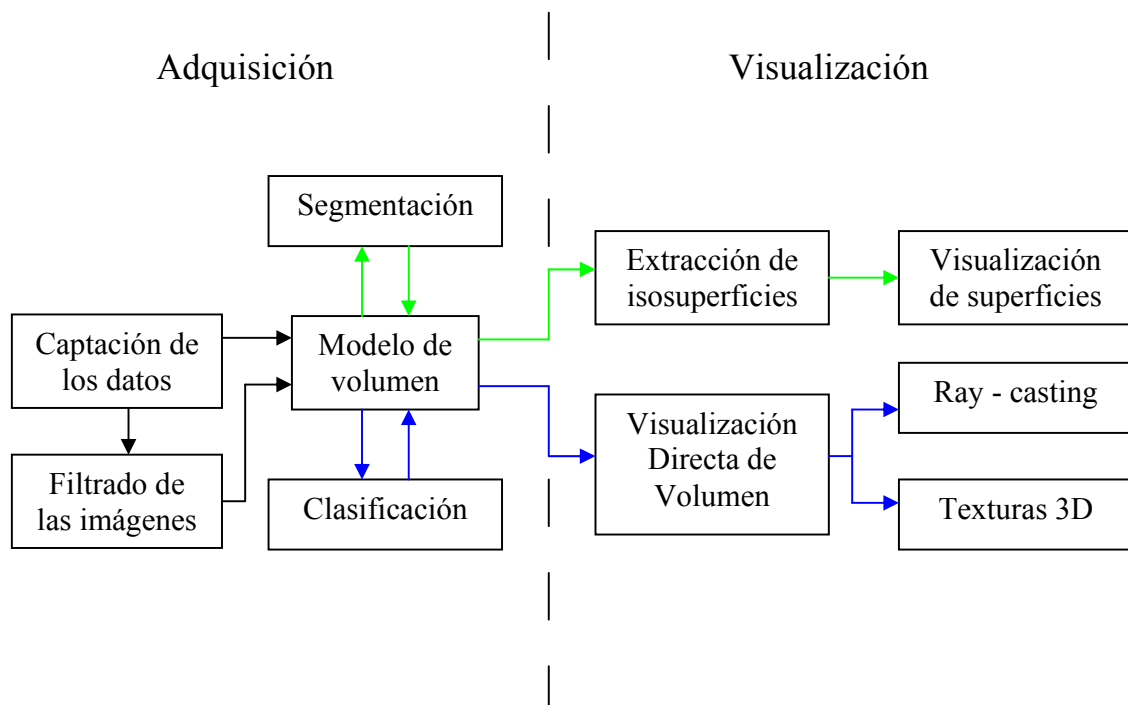


Figura 2.1: Etapas en la visualización de datos médicos.

2.2 Captación de datos médicos

El primer paso en cualquier sistema de visualización de estructuras anatómicas es la captación de los datos del paciente. Con este fin, el procedimiento habitual suele constar de dos fases. La primera de ellas es preparar al individuo para la adquisición de los datos, que dependerá del tipo de prueba que se le vaya a realizar. Este preproceso puede abarcar desde la inyección de un agente de contraste hasta la dilatación de determinados órganos con aire. Una vez que se ha llevado a cabo este preproceso, se introduce al paciente en un escáner para obtener las imágenes de su interior.

Los escáneres empleados para la adquisición de datos médicos permiten obtener, o bien un conjunto de imágenes 2D (ver figura 2.2) que representan cortes transversales del paciente (*slices*), o directamente un conjunto de *voxels* (ver apartado 2.3) que codifican la información de las estructuras anatómicas. Para mejorar la calidad de las imágenes obtenidas y eliminar posibles errores debidos al ruido del sistema de captación, es conveniente realizar un filtrado de los datos. A continuación se describen las modalidades más utilizadas para capturar imágenes médicas.

- *Tomografía computerizada (CT)*

La tomografía computerizada fue la primera técnica que permitió la adquisición de imágenes médicas. Esta modalidad de captación registra la impermeabilidad de las diferentes estructuras anatómicas a los rayos-X, mostrándose especialmente útil para conseguir imágenes de gran calidad de las estructuras óseas. Su principal inconveniente es que el paciente está expuesto a una dosis de radiación durante el proceso.

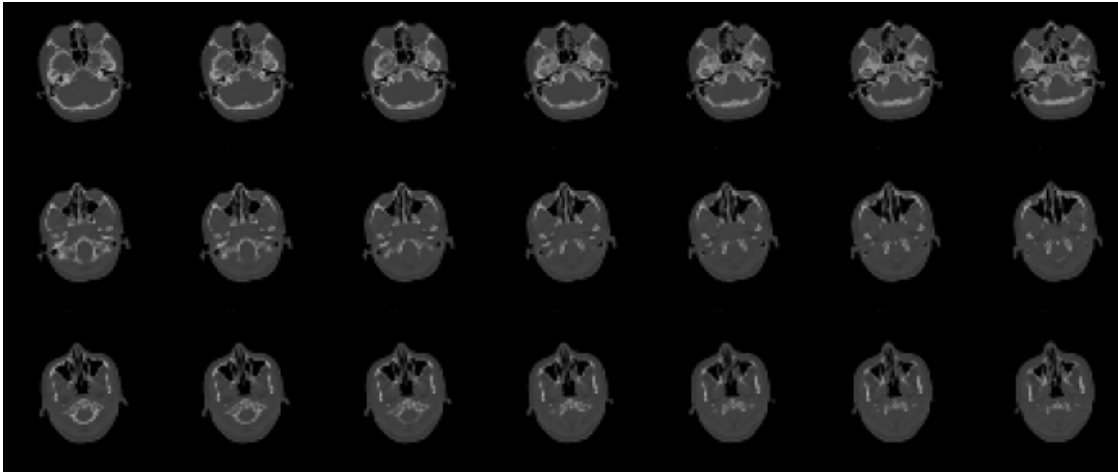


Figura 2.2: Imágenes CT del cráneo de un paciente.

- *Resonancia magnética (MRI)*

Las imágenes adquiridas mediante resonancia magnética son idóneas para conseguir información de tejidos blandos (que contienen agua). Basada en la alineación de los protones de hidrógeno al aplicar un campo magnético, se considera un método no invasivo ya que, a diferencia de las *CT*, el paciente no está expuesto a ningún tipo de radiación. A pesar de ello, las imágenes obtenidas son de menor calidad que las *CT* y se suele necesitar la presencia de un agente de contraste para mejorar el resultado.

- *Ultrasonidos*

Esta modalidad de adquisición de datos está basada en la emisión de ondas acústicas de alta frecuencia (entre 2 y 15 MHz) y en la recepción del eco producido por los tejidos a su paso. Los ultrasonidos son muy utilizados en algunos campos de la medicina, como por ejemplo para controlar la evolución del embarazo (ver figura 2.3) o para realizar ecocardiogramas.



Figura 2.3: Imagen de un embarazo obtenida con ultrasonidos.

- *Otras modalidades*

Existen otras técnicas para la adquisición de datos médicos, como por ejemplo la *Positron Emission Tomography (PET)* o la *Single Photon Emission Computer Tomography (SPECT)*, basadas en inyectar al paciente un isótopo radioactivo, con el fin de capturar la circulación de la sangre o la reacción que produce con algunas estructuras o tejidos (tumores, por ejemplo).

2.3 Modelo de volumen

Un modelo de volumen o volumétrico es aquel que permite almacenar la información muestreada en un conjunto de puntos distribuidos en un espacio 3D.

Los modelos de volumen son utilizados en muchas disciplinas ya que la información que son capaces de almacenar puede representar conceptos muy diferentes: unidades de temperatura, elevación del terreno, valores de densidad de cierto tejido, etc.

El esquema de representación más utilizado para datos distribuidos en mallas regulares es el modelo de *voxels*, que se puede entender como la extensión en 3D de la representación con imágenes. Usualmente se define una imagen como una estructura de datos discreta en 2D (tabla) constituida por píxeles. Cada píxel representa la mínima unidad de la que se compone la imagen y guarda un cierto valor. Análogamente, un modelo de *voxels* es una estructura de datos 3D donde cada *voxel* representa la mínima unidad de la que se compone el volumen y también guarda un valor. El espacio representado por un *voxel* es un cubo o un paralelepípedo.

En el campo de la medicina virtual, los modelos de volumen son muy útiles para almacenar la información obtenida en la etapa de adquisición de datos. Para ello, se considera que los puntos muestreados están distribuidos en una malla regular tridimensional, donde cada imagen contiene la información de un conjunto de puntos (i, j, k) de la malla con la componente k constante.

Para construir un modelo de *voxels* a partir de la captación, basta con formar una rejilla en tres dimensiones con todas las imágenes, de manera que en el centro de cada *voxel* se almacena una sola muestra. Al espacio que tiene como vértices un conjunto de muestras vecinas se le llama *celda* (ver figura 2.4). El valor de propiedad en puntos ubicados entre muestras se estima, usualmente por interpolación *trilineal*, a partir del valor de las muestras vecinas.

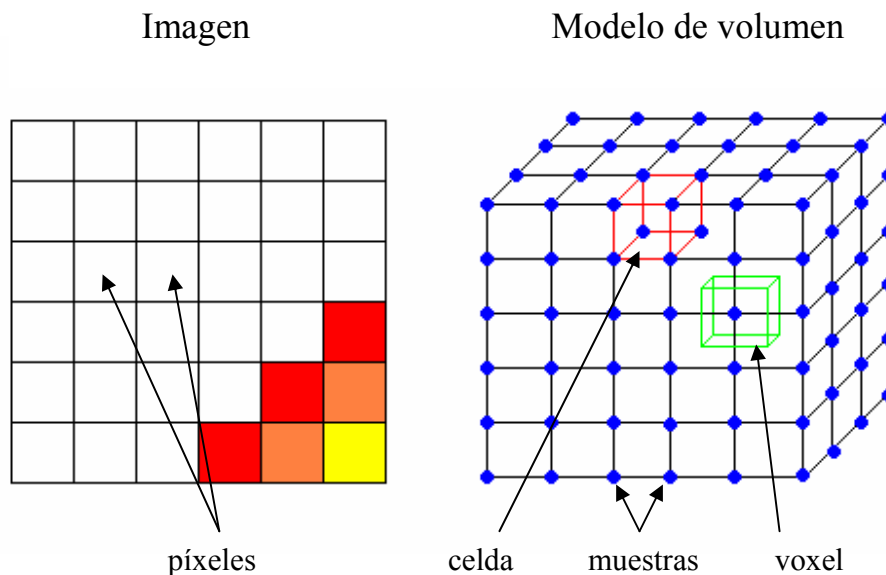


Figura 2.4: Esquema de una imagen y de un modelo de volumen con sus elementos.

El número de muestras en cada dimensión del volumen se denomina resolución. En las aplicaciones médicas la separación entre las muestras pertenecientes a una imagen es constante y relativamente pequeña (del orden de 0,1mm en los sistemas actuales); mientras que la separación entre muestras de imágenes diferentes (i, j, k) e $(i, j, k+1)$ acostumbra a ser mayor (como mínimo 1mm).

Existen otros esquemas de representación de volumen. Algunos de ellos, como los árboles octales, permiten representar la información de las mallas regulares de forma más compacta. Otros permiten guardar la información de mallas irregulares (puntos distribuidos de manera no uniforme) como las representaciones mediante tetraedros.

En esta tesis nos centraremos en la representación con modelos de *voxels* por ser la más utilizada en las aplicaciones médicas. En este sentido, cuando se hable de modelo de volumen se hará referencia a este tipo de representación, siempre que no se indique lo contrario.

2.4 Segmentación de los datos

El proceso de segmentación es una etapa fundamental en el análisis y procesado de imágenes y datos de volumen. El objetivo que persigue es el de aislar, extraer o identificar en los datos diferentes estructuras anatómicas. De su precisión y calidad dependerán los resultados obtenidos posteriormente en la etapa de visualización y de análisis.

A lo largo de los años se han desarrollado diferentes técnicas para llevar a cabo la segmentación. Algunas de ellas son manuales, otras semiautomáticas, y algunas totalmente automatizadas, aunque generalmente éstas últimas presentan un alto coste computacional. Recientemente, aprovechando la velocidad de cálculo de los procesadores gráficos, se están implementando técnicas de segmentación que permiten obtener mejores resultados en menos tiempo [6].

A gran escala, podemos clasificar las técnicas de segmentación en tres grandes grupos:

- *Métodos basados en los píxeles de la imagen (Pixel-based methods)*

Estas técnicas trabajan directamente sobre los píxeles de las imágenes clasificándolos según un cierto criterio. La técnica conocida como *thresholding* pertenece a esta familia y consiste en incluir o excluir los *voxels* del volumen según si su valor de propiedad es superior o inferior a un determinado límite (*threshold*).

- *Métodos basados en los límites de las estructuras (Edge-based methods)*

Estos métodos utilizan funciones de mínimo coste e información de los límites de las estructuras para determinar los bordes de las regiones homogéneas.

Algunas técnicas de esta familia utilizan filtros, normalmente basados en el valor del gradiente, para detectar los contornos. Cuando el resultado no proporciona elementos cerrados se aplican técnicas adicionales para conectarlos (operadores morfológicos, por ejemplo).

- *Métodos basados en regiones de la imagen (Region-based methods)*

Estas técnicas intentan reconstruir las diferentes estructuras anatómicas utilizando un determinado criterio de similitud, que se ha de cumplir en los píxeles de cada estructura. Uno de los métodos de esta familia es el *region growing*, que parte de una o varias regiones iniciales llamadas *semilla*, determinadas de forma manual o automática mediante un heurístico. Las regiones se van expandiendo a los *voxels* vecinos mientras se cumple un determinado criterio de homogeneidad.

En la figura 2.5 se puede ver el resultado de aplicar *region growing* sobre imágenes *CT* del abdomen. En este ejemplo las *semillas* se han colocado manualmente en cada una de las regiones coloreadas, que se han ido expandiendo de forma automática por los *voxels* vecinos con un valor de propiedad parecido al de las *semillas*.

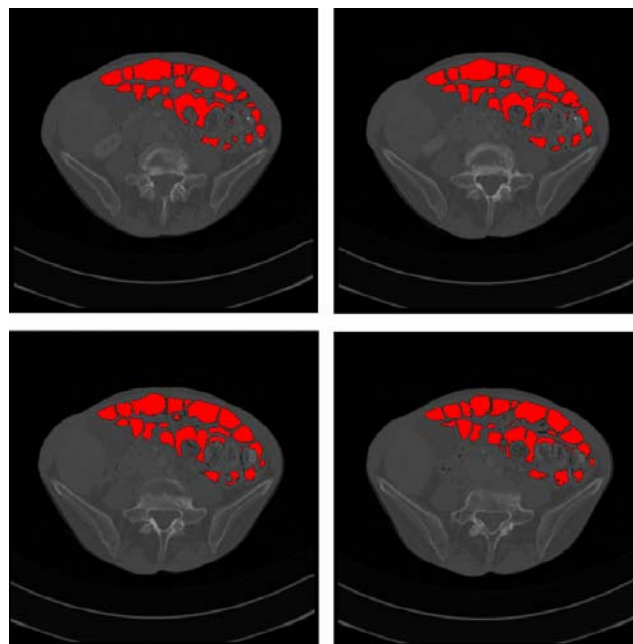


Figura 2.5: Segmentación de los intestinos con region-growing a partir de imágenes CT.

Otros métodos de esta familia son los *shape-based approaches*, que mantienen en todo momento una representación de la forma del elemento a segmentar. Estas formas, también conocidas como *active shape models* o *active contour models*, se pueden modificar manualmente para adaptarlas mejor a un determinado elemento, o pueden ajustarse de forma automática según el valor de una cierta función de energía, como se realiza en el ejemplo de la figura 2.6.

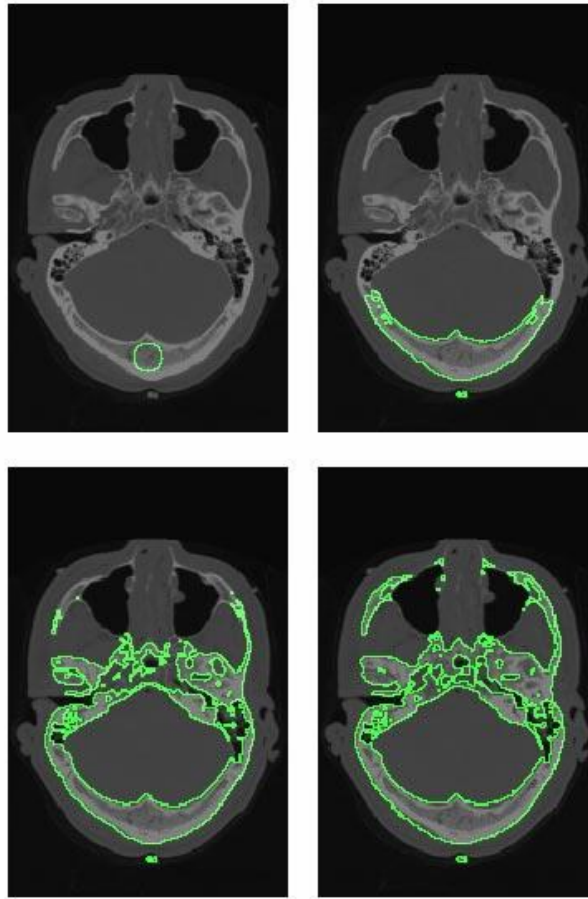


Figura 2.6: Segmentación automática de las estructuras óseas del cráneo utilizando shape-based approaches. Imagen extraída de [22].

2.5 Clasificación de los datos

Según la técnica que se utiliza para realizar la visualización del modelo de volumen, no siempre es necesario pasar por la etapa de segmentación. Hay veces en las que basta con realizar una clasificación de los *voxels* según el valor que almacenan. El objetivo de este proceso no es el de extraer las estructuras del volumen, sino el de identificarlas en función de sus valores de propiedad. Así, en la posterior visualización es posible observar todos los datos del volumen, no tan solo aquellos pertenecientes a los órganos segmentados. Además de ser un proceso alternativo, la clasificación también se puede realizar como paso previo para aumentar la precisión en la segmentación.

Para llevar a cabo la clasificación se define lo que se conoce como *función de transferencia* [3], encargada de asignar un color *RGBA* a cada valor de la propiedad. Usualmente, la opacidad *A* diferencia las estructuras anatómicas que se encuentran en el volumen y los componentes *RGB* indican sus propiedades ópticas.

Una forma sencilla de definir la función de transferencia es utilizar como base el histograma del volumen, que aporta información sobre el comportamiento de los datos en el dominio de la función. De esta manera, asignando valores *RGBA* a los diferentes rangos del histograma se pueden identificar fácilmente los elementos del volumen. En ocasiones los rangos de valores del histograma no están del todo definidos y no se puede saber con precisión cuales son los límites de cada elemento (*fuzzy regions*), para solucionar este problema existen métodos más complejos que permiten realizar la clasificación con mayor precisión [3].

En la figura 2.7 se muestran dos ejemplos de funciones de transferencia basadas en el histograma del volumen. El eje X codifica el valor de propiedad de los datos, el eje Y codifica la opacidad y a cada rango de valores se le asigna un color diferente (tres rangos de color rojo, naranja y blanco en la función de la izquierda y dos rangos, rojo y beige en la función de la derecha). Obsérvese que la opacidad en los límites de cada uno de los rangos aumenta o decrece progresivamente, pero no se mantiene constante como en el resto del rango. De esta manera se pueden disimular las imprecisiones producidas por las zonas *fuzzy* del histograma en la visualización final.



Figura 2.7: Ejemplo de funciones de transferencia.

2.6 Visualización de los datos

Las técnicas encargadas de la visualización se ocupan de transformar los datos adquiridos del paciente en las imágenes finales que analiza el especialista. El objetivo principal de esta etapa no es solo conseguir imágenes de calidad, sino también obtenerlas con la mayor rapidez y precisión posible, de forma que se pueda llevar a cabo una inspección interactiva.

A lo largo de los años se han propuesto diferentes métodos para realizar la visualización de datos de volumen, que de forma muy general se pueden clasificar en dos categorías: Visualización de isosuperficies y Visualización Directa del Volumen (*Direct Volume Rendering* o *DVR*).

2.6.1 Visualización de isosuperficies

Estas técnicas se basan en pintar una estructura geométrica, generalmente una malla de triángulos, que representa la superficie de las estructuras anatómicas del volumen. Existen diferentes métodos que permiten reconstruir las superficies a partir de los datos segmentados; el más utilizado es el algoritmo de *Marching Cubes* propuesto por Lorensen y Cline en 1987 [14].

Una isosuperficie se define como el conjunto de puntos presentes en el volumen de datos con un mismo valor de la propiedad. El algoritmo de *Marching Cubes* (*MC*) permite extraer una malla poligonal que reconstruye una isosuperficie.

El primer paso del algoritmo consiste en etiquetar cada muestra del volumen según si su valor es superior o inferior al de la isosuperficie que se quiere reconstruir. Así, las muestras quedan clasificadas en dos estados, dentro o fuera de la superficie. A continuación, se crea la malla de triángulos en aquellas celdas por las que pasa la isosuperficie, es decir, las celdas que tienen vértices de los dos tipos.

El valor del gradiente en una muestra coincide con el vector normal a la isosuperficie que la contiene. Con esta premisa, se obtienen por interpolación lineal en las aristas con vértices con diferente clasificación, los puntos con el valor de la isosuperficie y sus normales. Estos puntos son los vértices de los triángulos de la malla resultante. Una de las principales ventajas del algoritmo de *MC* es que permite reconstruir la isosuperficie de forma local (en cada celda).

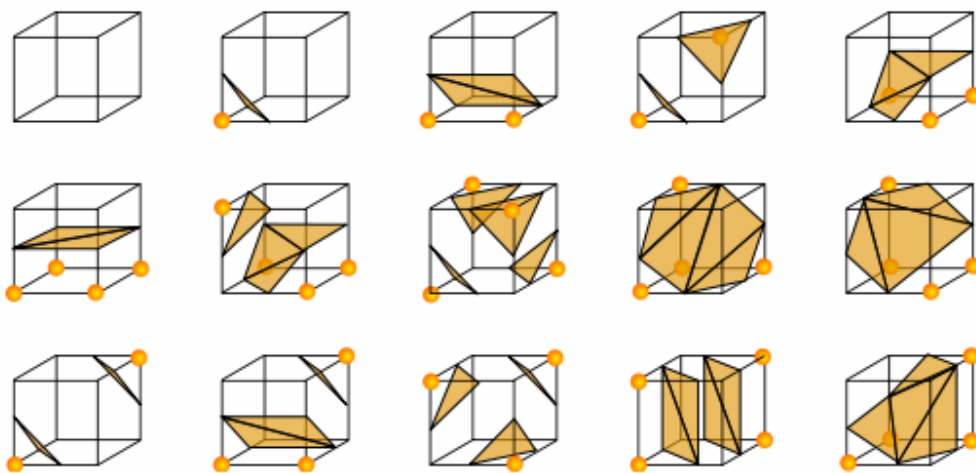


Figura 2.8: Las 15 configuraciones del algoritmo de *Marching Cubes*.

Como cada celda tiene 8 vértices (muestras del volumen) que pueden ser clasificados en dos posibles estados (dentro o fuera de la superficie), existen 256 tipos diferentes de celdas, que se pueden reducir a las 15 configuraciones mostradas en la figura 2.8. El problema principal de la técnica de *Marching Cubes* es que algunas de las configuraciones son ambiguas, es decir, la isosuperficie se puede reconstruir de varias maneras dentro de una misma celda.

Desde que se presentó esta técnica en 1987 se han propuesto diferentes versiones para resolver las ambigüedades, como por ejemplo dividir las celdas en tetraedros antes de realizar la reconstrucción (*marching tetrahedrons*). Aplicando el algoritmo de *marching cubes* a los tetraedros en vez de a las celdas no se obtienen configuraciones ambiguas.

La exactitud de la malla de triángulos obtenida para cada isosuperficie depende de los resultados obtenidos en la segmentación. Si las estructuras anatómicas del volumen no están bien segmentadas, se pueden producir solapamientos incorrectos entre ellas o se puede generar geometría en zonas donde no debería haberla.

La principal ventaja que presenta la visualización de isosuperficies respecto a las técnicas de *Direct Volume Rendering*, es que permite obtener las imágenes en menor tiempo. Por otra parte, una de las limitaciones de este método es que el número de polígonos que se pueden llegar a generar para volúmenes relativamente pequeños es muy elevado (millones de triángulos), por lo que cuesta obtener *frame rates* interactivos en grandes volúmenes de alta resolución.

Para reducir el número de triángulos que se envían a pintar, se puede actuar de dos maneras: simplificar la malla reconstruida o pintar solo aquellas partes que sean de interés. El principal inconveniente de simplificar la malla es que va perdiendo fidelidad respecto a la isosuperficie original, por lo que se pueden eliminar características significativas de las superficies reconstruidas.

2.6.2 Visualización Directa del Volumen (DVR)

Otra forma de visualizar datos de volumen es lo que se conoce por *DVR*, donde no se reconstruye ninguna estructura geométrica y permite visualizar toda la información contenida en el modelo de volumen.

Las técnicas basadas en *DVR* consideran el volumen de datos como una distribución de partículas emisoras de luz. Para realizar los cálculos que intervienen en la iluminación se pueden considerar diferentes modelos ópticos [15]: tener en cuenta solo la absorción de las partículas, la emisión, ambas, la transmisión y reflexión, el sombreado, etc.

El modelo óptico más utilizado en estas técnicas es el que tiene en cuenta la absorción y emisión de las partículas. Hay dos formas de llevar a cabo la composición del color: de delante hacia atrás (*front-to-back*) o al revés (*back-to-front*) respecto al observador. Mientras que la segunda es más sencilla, la primera tiene la ventaja de terminar la composición cuando se alcanza un valor de opacidad determinado.

Desde que se presentaron los primeros algoritmos para *DVR* [13] hasta la actualidad, se han utilizado diferentes técnicas para su implementación. A gran escala se pueden clasificar en dos grupos: los métodos que utilizan lo que se denomina como *Image-order approach*, basados en trazar rayos desde cada píxel de la imagen hasta el volumen, y los que usan el *Object-order approach*, basados en proyectar el volumen sobre el plano de la imagen.

Una de las características más relevantes de estos métodos es que se obtienen resultados de mucha calidad, pero el tiempo que invierten en realizar los cálculos es elevado. Por este motivo, se han desarrollado un conjunto de técnicas destinadas a acelerar este sistema de visualización, que se comentan brevemente en los siguientes apartados de este capítulo.

A continuación se presentan las dos técnicas más utilizadas en *DVR*: *ray-casting* y el uso de texturas.

- *Ray-casting*

Esta técnica que sigue el *Image-order approach* se basa en trazar un rayo desde el observador hasta el centro de cada píxel, calcular el color y la opacidad de las muestras del volumen atravesadas por el rayo, y realizar la composición del color teniendo en cuenta la opacidad de las muestras (*alpha blending*).

Utilizando *ray-casting* se obtienen imágenes de mayor calidad que con la visualización de isosuperficies o con texturas (ver figura 2.13). A pesar de ello, la mayor limitación de esta técnica es el tiempo que tarda en obtener las imágenes finales, que puede ser de varios minutos dependiendo de la resolución de la imagen y del tamaño del volumen.

A lo largo de los años se han presentado numerosas técnicas para reducir el tiempo de cálculo empleado por este método. La conocida como *early ray termination* se basa en acabar de recorrer cada rayo cuando se alcanza la opacidad máxima en la composición de color. Para que funcione correctamente la composición se ha de realizar en orden *front-to-back*. Otra forma de acelerar los cálculos es la técnica llamada *empty-space skipping* que no recorre los *voxels* vacíos del volumen, porque no aportan información relevante para el cálculo del color. Kruger y Westermann han desarrollado dos variantes de estas técnicas implementadas con la *GPU* [10].

Algunos de los sistemas presentados en el capítulo 3 utilizan otras técnicas de aceleración como el *surface-assistant ray-casting* [21] o el cálculo en paralelo de las operaciones para la composición de color.

- *Uso de texturas*

Esta modalidad de visualización pertenece al *Object-order approach* y consiste en pintar polígonos texturados con la información del volumen. Normalmente la geometría que se envía a pintar son un conjunto de planos y se obtienen resultados de calidad parecida a los del *ray-casting* (ver figura 2.13). Para visualizar el modelo existen dos posibilidades: utilizar un conjunto de texturas 2D o una textura 3D.

La manera más directa para visualizar el modelo de volumen es guardar las *slices* de la captación en texturas 2D y enviar a pintar un plano texturado por cada una de ellas. Como resultado se obtiene un conjunto de planos alineados con el volumen (ver figura 2.9 y parte superior de figura 2.10). La principal limitación de esta técnica (como se observa en la figura 2.9) es que para determinados puntos de vista existen zonas de la imagen en las que no se proyecta ninguna textura, con lo que se puede observar la separación entre planos. Para solucionar este problema, se calculan y almacenan tres conjuntos de *slices*, cada uno perpendicular a un eje coordenado, y se visualiza siempre el conjunto más cercano a la perpendicularidad con la dirección de visión.

Además de tener codificado el volumen por triplicado, otro de los inconvenientes del uso de texturas 2D es que el número de planos con los que se visualiza el volumen queda determinado por el número de *slices* de la captación. Esto provoca que los errores debidos a la frecuencia de muestreo no se puedan eliminar. En el capítulo 4 veremos que una forma de eliminar estos *artifacts* es aumentar el número de planos.

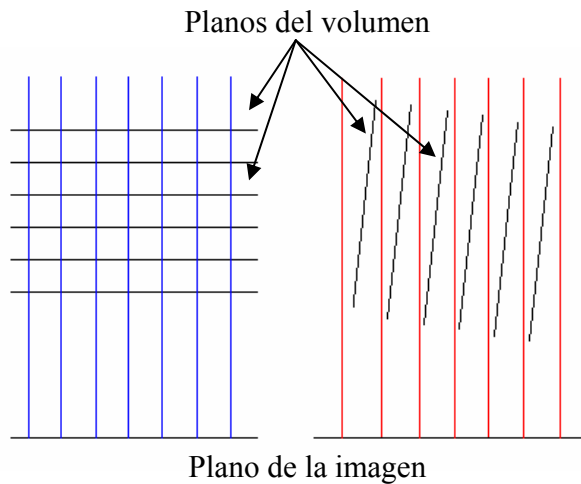


Figura 2.9: Dependiendo del punto de vista puede haber zonas en el plano de la imagen donde no se proyecta el volumen (líneas rojas del caso de la derecha).

Una alternativa al método anterior es construir una textura 3D con el modelo de volumen, guardando en cada elemento de la textura (*texel*) el valor de propiedad de un *voxel*. En el momento de la visualización se envían a pintar un conjunto de planos perpendiculares a la dirección de visión, a los que se asigna las coordenadas de la textura 3D correspondientes a su posición en el volumen muestreado.

Con esta modalidad, los planos pueden permanecer perpendiculares a la dirección de visión (ver figura 2.10), por lo que no hace falta triplicar el volumen. Con este método resulta muy sencillo aumentar el número de planos que se envían a pintar, lo que permite disminuir la distancia de muestreo y aumentar la calidad de las imágenes.

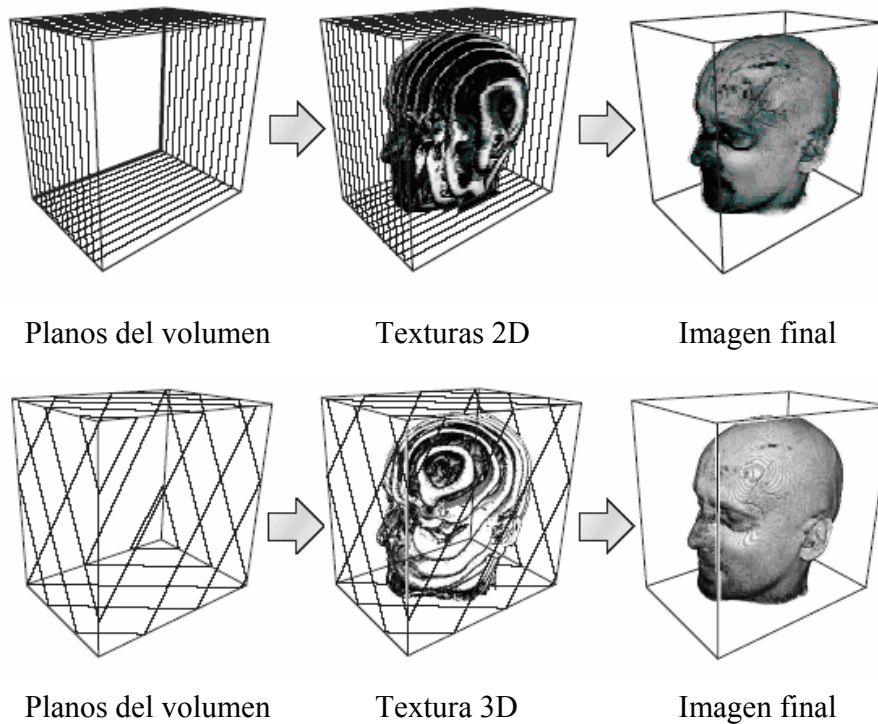


Figura 2.10: Esquema de planos alineados con el objeto utilizando texturas 2D (arriba) y planos ortogonales a la dirección de visión usando una textura 3D (abajo). Imagen extraída de [3].

Para calcular la opacidad en la composición del color se tiene en cuenta, además de los valores que se obtienen de la clasificación, la distancia entre muestras. Este cálculo es más sencillo si la distancia entre muestras se mantiene constante.

La ventaja principal que presenta el uso de texturas es que permite obtener imágenes de gran calidad en menos tiempo que el método de *ray-casting*, con lo que fácilmente se pueden conseguir *frame rates* interactivos, sobretodo si se utiliza el hardware gráfico para manejar la texturación.

Como se ha comentado anteriormente y se verá con detalle en el capítulo 4, uno de los problemas de esta técnica es que la calidad de la imagen obtenida está directamente relacionada con la frecuencia de muestreo, de manera que cuando se envía a pintar el volumen con pocos planos se detectan una serie de errores en el resultado (ver figura 2.11).

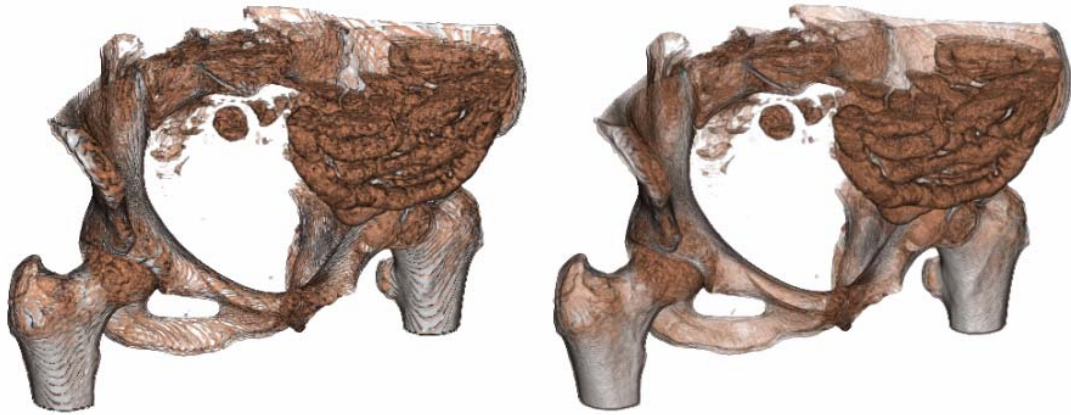


Figura 2.11: Modelo de volumen de $512 \times 512 \times 171$ voxels visualizado con texturas 3D, usando 200 (izq.) y 500 planos (der.).

Como se ha indicado, cuando la distancia entre planos es constante, para el cálculo de la opacidad se considera una atenuación igualmente constante. Esta hipótesis es cierta cuando se utiliza la proyección axonométrica para visualizar el volumen. Hay ocasiones, como cuando se inspecciona el interior de los órganos, en las que es preferible disponer de una vista en perspectiva, ya que permite obtener la sensación de profundidad que no se consigue con la axonométrica. En este caso, la distancia de muestreo de planos paralelos no es la misma para todos los píxeles en que se proyectan (ver figura 2.12), lo que puede provocar errores al realizar la composición de color. Una de las propuestas para solucionar este problema es utilizar geometría curvada (*spherical shells*) en lugar de planos para pintar el volumen. Estas *shells* son esferas concéntricas recortadas por el *frustum* de visión y por la caja contenedora del modelo de volumen. Los principales inconvenientes de esta propuesta son que se pintan más polígonos que con los planos y que los cálculos para la texturación resultan más complejos.

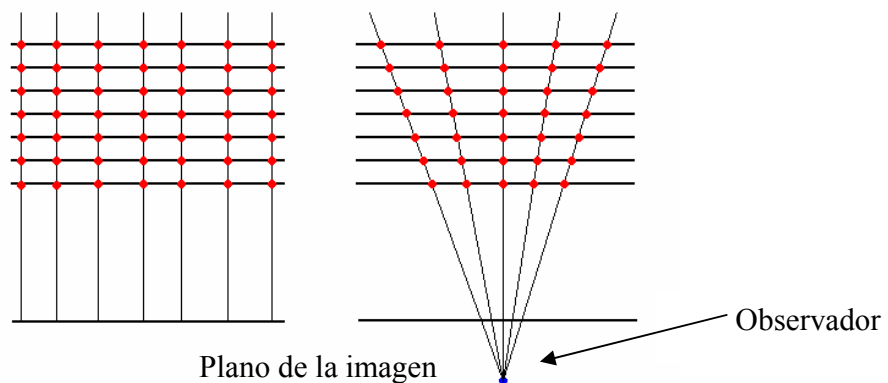


Figura 2.12: Distancia de muestreo existente con la proyección axonométrica (izq.) y con la proyección perspectiva (der.).

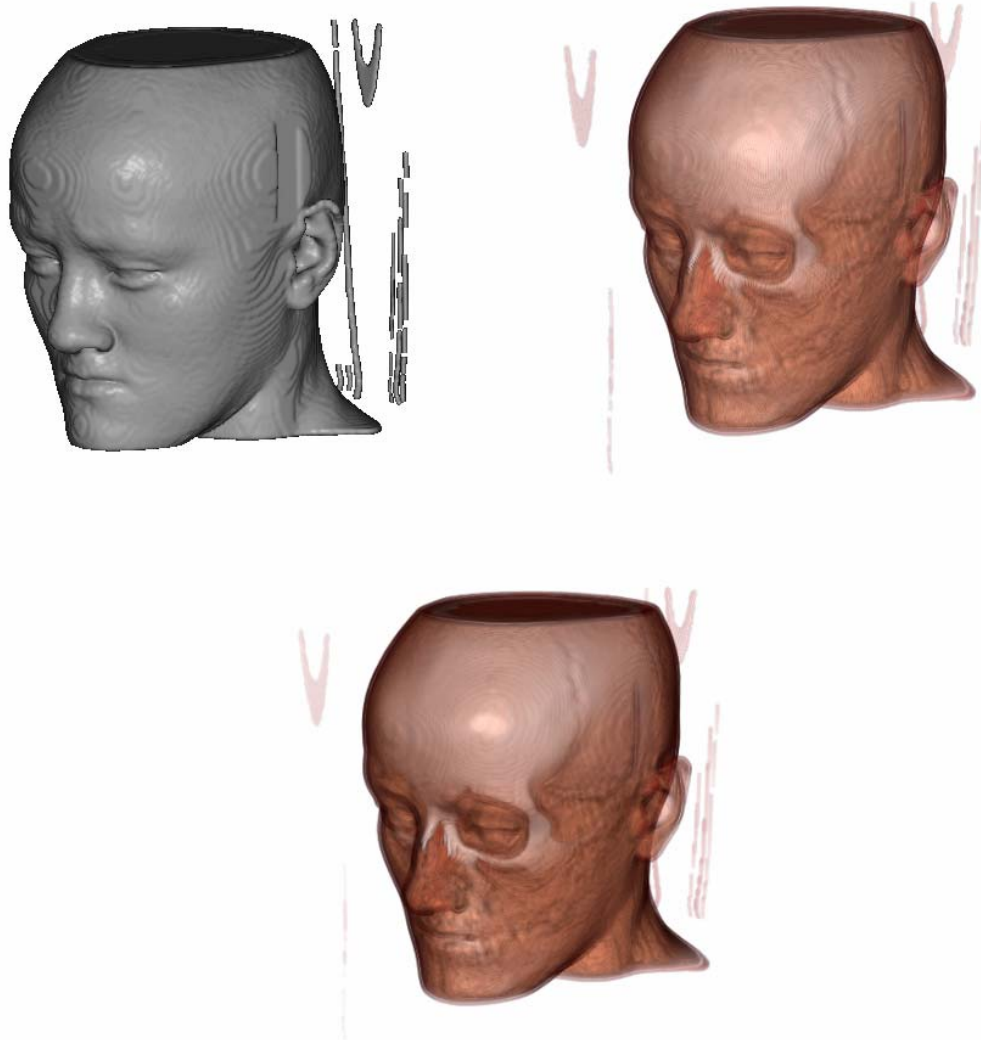


Figura 2.13: Imágenes obtenidas con visualización de isosuperficies (arriba, izquierda), ray-casting (arriba, derecha) y texturas 3D (abajo).

2.7 Etapas de una endoscopia virtual

En los apartados anteriores de este capítulo se ha presentado una descripción general de las etapas que intervienen en la visualización de datos médicos. Para poder realizar una endoscopia virtual hay que tener en cuenta otros procesos además de la adquisición y la visualización, como por ejemplo el tratamiento del paciente antes de la captación o la forma en la que el usuario inspeccionará los órganos visualizados (ver figura 2.14).

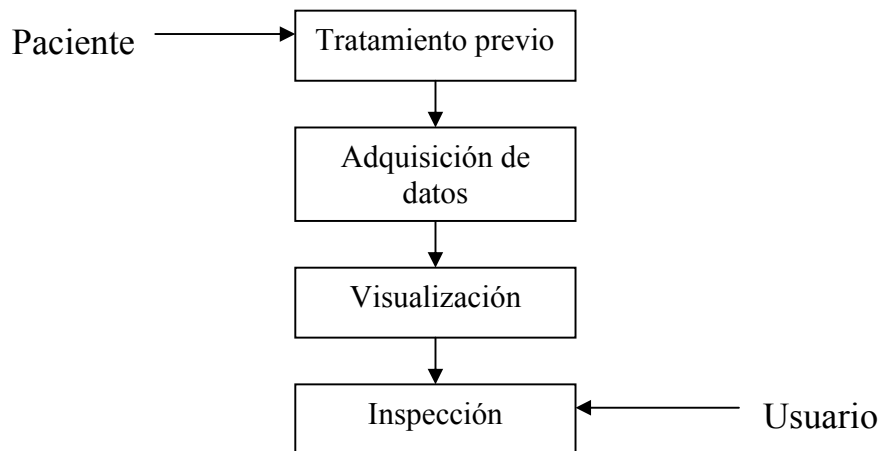


Figura 2.14: Etapas de una endoscopia virtual.

- *Tratamiento previo*

Para aumentar la calidad de las imágenes y facilitar la segmentación de las estructuras tubulares es necesario someter al paciente a un tratamiento previo. Antes de realizar una colonoscopia virtual, por ejemplo, se han de llevar a cabo las mismas operaciones de limpieza del colon que se efectúan al practicar una colonoscopia convencional. A continuación se suele introducir aire en el intestino del paciente para dilatarlo de manera que se pueda capturar con mayor precisión el conducto (no queden puntos ocultos en pliegues del colon, etc.).

Otro de los procesos habituales al que se puede someter al paciente es la ingestión o inyección de un agente de contraste para identificar con claridad el conducto respecto a sus paredes.

- *Adquisición y visualización*

Para las endoscopias virtuales se suelen utilizar datos procedentes de *CT* o *MRI*. La adquisición mediante ultrasonidos no se utiliza en estas aplicaciones ya que las imágenes obtenidas no tienen suficiente calidad (presentan mucho ruido). Tampoco son apropiadas otras técnicas como la *PET*, la *SPECT*, la *fluoroscopia* o las radiografías convencionales.

Para la visualización de los datos se pueden utilizar las diferentes técnicas presentadas en este capítulo. Algunos de los sistemas que se han desarrollado hasta la fecha se basan en la extracción de isosuperficies y pintado de mallas poligonales, otros utilizan las técnicas de *ray-casting*, texturas o incluso hay sistemas que ofrecen métodos híbridos entre *DVR* y mallas poligonales. En el capítulo 3 se presenta una descripción más detallada de los sistemas existentes.

- *Inspección*

Una vez el sistema nos muestra el interior del paciente, fruto de los resultados conseguidos en las etapas anteriores, se le han de proporcionar al especialista las herramientas para que inspeccione los órganos visualizados. Hasta la fecha se han propuesto tres modalidades diferentes de navegación:

- *Navegación automática*: la exploración se lleva a cabo mediante una animación generada automáticamente por el sistema. El recorrido por el interior del conducto a explorar se puede generar de forma automática o lo puede definir el usuario marcando alguno de los puntos por los que pasará la cámara.
- *Navegación libre o manual*: en este tipo de navegación el usuario tiene control total de la cámara, tanto la posición como la orientación. Incorpora un sistema de control de colisiones para que la cámara no abandone el conducto inspeccionado.
- *Navegación guiada*: el usuario controla la orientación de una cámara virtual que sigue un camino preestablecido por el interior de las estructuras anatómicas.

2.8 Resumen

En este capítulo se han comentado las etapas básicas de las que consta un sistema de visualización de datos médicos, se han descrito las técnicas utilizadas en cada etapa y se ha terminado describiendo las etapas específicas de una endoscopia virtual. En el capítulo 3 veremos cuales de las técnicas explicadas se han utilizado en los sistemas de endoscopia virtual presentados hasta la fecha, así como las limitaciones existentes.

Capítulo 3

Estado del arte en endoscopia virtual

3.1 Introducción

Desde la presentación de los primeros sistemas, se han desarrollado múltiples aplicaciones destinadas a la inspección virtual de estructuras anatómicas tubulares (endoscopia virtual). Algunas de ellas son fruto del trabajo científico presentado en revistas especializadas y otras son sistemas comerciales. En este capítulo se pretende describir algunos de los sistemas existentes y mostrar sus características principales.

3.2 Sistemas no comerciales

Entre los grupos de investigación que se han interesado por los avances y técnicas relacionadas con las endoscopias virtuales, destacan nombres propios como Dirk Bartz, Rüdiger Westermann o William E. Lorensen. En colaboración con algunas universidades y centros médicos, otros muchos investigadores han desarrollado sistemas como los que se presentan a continuación.

- *FreeFlight de la Universidad de Wake Forest*

Presentado por un grupo de investigadores de la universidad de *Wake Forest*, *FreeFlight* [19] es uno de los primeros sistemas dedicados a realizar endoscopias virtuales. Utilizando datos captados mediante *CT* y *MRI*, segmenta el volumen aplicando las técnicas de *thresholding* y *region-growing*. A continuación reconstruye las isosuperficies con el algoritmo de *Marching Cubes*.

Ofrece un sistema de navegación manual utilizando las técnicas de interacción de la *OpenInventor API* [16], además de otro sistema de navegación automática. El principal inconveniente de esta aplicación es la dificultad de uso de la navegación manual, poco intuitiva para los usuarios.

- *VICON de la State University of NY at Stony Brook*

En 1994 se empezó a desarrollar este sistema pensado para realizar colonoscopias virtuales. Utilizando una modalidad de *region-growing* en la etapa de segmentación, se obtiene una malla poligonal con la técnica de *Marching Cubes* para visualizar las estructuras del volumen.

Con los años se han ido incorporando mejoras en el sistema, como la implementación de un módulo de *DVR* mediante *ray-casting* para aumentar la calidad de las imágenes resultantes.

Inicialmente, el sistema *VICON* [8] implementaba una modalidad de navegación automática donde la cámara seguía un camino precalculado durante la segmentación, aplicando la técnica de *onion peeling* [4] al modelo de volumen. A continuación se generaba una animación *offline* de mayor calidad con el módulo de *DVR*.

Posteriormente se implementó un sistema de navegación guiado basado en lo que se conoce como *submarine model* [7]. El usuario podía controlar una cámara sujeta a un conjunto de ecuaciones cinemáticas extraídas de las fuerzas existentes en una exploración convencional, lo que permitía evitar colisiones con las paredes del intestino.

- *VIVENDI de la Universidad de Tübingen*

Basado en el sistema *VICON*, el sistema *VIVENDI* [2] lleva a cabo la visualización a partir de mallas poligonales obtenidas mediante el algoritmo de *Marching Cubes*. Utilizando datos de entrada captados mediante *CT*, *MRI*, *Rotacional Angiography (C-Arm Xray)* u otros escáneres 3D, se implementa la etapa de segmentación con una técnica de *region-growing* en la que el usuario ha de especificar una semilla y un cierto *threshold*.

Para la inspección de las diferentes estructuras anatómicas se implementa el mismo sistema de navegación guiada que utiliza *VICON*. El camino por defecto que sigue la cámara virtual se genera al mismo tiempo que se realiza la segmentación de los datos, utilizando el algoritmo de camino mínimo de *Dijkstra* [7].

- *VirEn de la Technical University of Vienna*

El sistema *VirEn* [18] permite aplicar a los datos obtenidos con *CT* o *MRI* algunas técnicas de procesado de imágenes para mejorar su calidad. Presenta un sistema de segmentación semiautomático donde el usuario debe introducir un cierto *threshold* o una semilla según la técnica que se vaya a aplicar. Una vez se tiene el volumen segmentado, se utiliza el algoritmo de *Marching Cubes* para obtener la malla de triángulos que se va a visualizar. También ofrece la posibilidad de pintar vistas concretas de mayor calidad implementando una variante de baja resolución de *ray-casting* para obtener *frame rates* interactivos.

Esta aplicación ofrece al usuario un sistema de navegación guiado donde la cámara va siguiendo un camino precalculado por el interior del órgano a inspeccionar. El camino óptimo se consigue mediante un proceso de esqueletización del volumen (*topological thinning*), donde cada *voxel* ha sido marcado como interior o exterior a la estructura anatómica explorada en el proceso de segmentación.

- *CRS4 en Cagliari, Italia*

El sistema *CRS4* [5] permite visualizar los datos de volumen mediante texturas 3D sin necesidad de realizar ningún tipo de segmentación. Para la clasificación de los datos se utilizan *look-up tables* que codifican las propiedades ópticas de los diferentes valores del volumen. El sistema incorpora un *ray-caster* 2D para obtener vistas concretas de mayor calidad.

Para adaptar la distancia entre las *slices* del volumen a la proyección perspectiva, el espacio entre planos se define inversamente proporcional a la distancia respecto al observador. De esta forma, la contribución de cada *slice* al resultado final disminuye a mayor distancia del observador y por tanto, se consigue más calidad en las zonas cercanas a la cámara y menos en las más alejadas. A pesar de ello, al no aplicar iluminación (*shading*) a las superficies visualizadas por cuestiones de rendimiento, se pueden distinguir los planos en las imágenes finales (ver figura 3.1).

Se implementa una técnica de navegación manual basada en la simulación física de las fuerzas que pueden actuar sobre la cámara dentro del conducto a explorar. También se ofrece al usuario una herramienta de *point picking* para manejar los widgets de la aplicación y medir características morfológicas de la zona visualizada (ver figura 3.1).

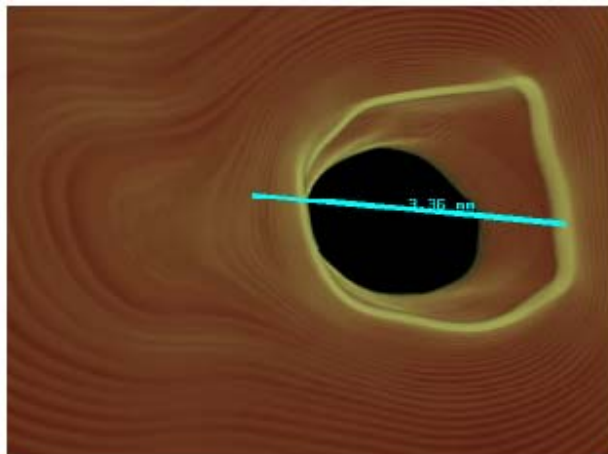


Figura 3.1: Ejemplo de medición efectuada con el sistema CRS4 en una arteria visualizada con texturas 3D. Imagen extraída de [5].

- *3D Slicer del MIT y del Brigham's and Women's Hospital*

3D Slicer [4] es un sistema basado en la visualización de mallas poligonales extraídas con la técnica de *Marching Cubes*. Presentado en 2001 como fruto de la colaboración entre investigadores del *MIT* y del *Brigham's and Women's Hospital* de Boston, utiliza las técnicas de *thresholding* y *boundary localization* para el proceso de segmentación.

Presenta tres modalidades diferentes de navegación, dos de ellas automáticas y una manual. El usuario define una serie de *key frames* para uno de los métodos automáticos, y el sistema se encarga de generar el resto de la animación por interpolación. En el segundo método de navegación automática la cámara sigue el camino central del conducto a explorar. Para la navegación manual, el control de colisiones se calcula según la distancia de la cámara a la superficie en la dirección de visión. Cuando la distancia es muy pequeña, se le aplican a la cámara unas fuerzas reactivas y se evita la colisión.

3.3 Sistemas comerciales

Algunas multinacionales han comercializado sistemas con los que realizar endoscopias virtuales. Es el caso de Siemens o Philips con los sistemas *Syngo* y *EasyVision Endo3D*. Otras empresas del sector también han desarrollado sus propias aplicaciones como es el caso de *Viatronix*, *Vital Images* o *GE Corporation*. A continuación se describen las características principales de algunos sistemas comerciales desarrollados por estas empresas.

- *VESA de GE Corporation*

Esta aplicación utiliza mallas poligonales para visualizar los datos anatómicos del paciente. Al conjunto de imágenes obtenidas mediante *CT* o *MRI* se les aplican los operadores morfológicos de *dilatación* y *erosión* para mejorar su calidad. A partir de ahí, se dispone de varias modalidades para realizar la segmentación de los datos y se reconstruyen las isosuperficies con el algoritmo de *Marching Cubes*.

Para la inspección de los órganos del paciente, el sistema ofrece dos modalidades diferentes de navegación: una de ellas manual y la otra automática. En la navegación manual, el usuario controla el movimiento de una cámara sujeta al conjunto de fuerzas que experimentaría en una navegación real. El sistema de navegación automática ofrece dos tipos de interacción al usuario. El primero permite seleccionar diferentes *key frames* y obtiene por interpolación el resto de la animación, mientras que el segundo define de forma automática el camino que sigue la cámara mediante un algoritmo de *robot path-planning* [12].

El sistema *VESA* [9] fue una de las primeras aplicaciones para realizar endoscopias virtuales que salieron al mercado. Actualmente *GE Corporation* ofrece el *Advantage Windows System* que utiliza la técnica de *ray-casting* para visualizar los datos de volumen.

- *V3D Viewer de Viatronix*

Utilizando la tecnología del sistema *VICON* explicado la sección 3.2, *Viatronix* ha desarrollado el sistema *V3D* [1] que ofrece al usuario la posibilidad de realizar múltiples operaciones, desde colonoscopias virtuales hasta explorar el sistema vascular.

A partir de los datos de entrada obtenidos por *CT* o *MRI* se ejecuta de forma automática la etapa de segmentación. Basándose en un algoritmo propio para clasificar los diferentes tejidos del volumen y en un proceso de *region-growing* para extraer las diferentes estructuras anatómicas, este sistema permite visualizar los datos pintando mallas de polígonos o directamente mediante *DVR*.

La aplicación proporciona al usuario un sistema de navegación guiada que sigue un camino precalculado automáticamente. Para ello calcula la línea central del conducto a explorar y el conjunto de fuerzas a las que está sometida la cámara. Con *V3D Viewer* se consiguen unos tiempos de 15 *frames* por segundo (*fps*), lo que permite una inspección fluida del volumen.

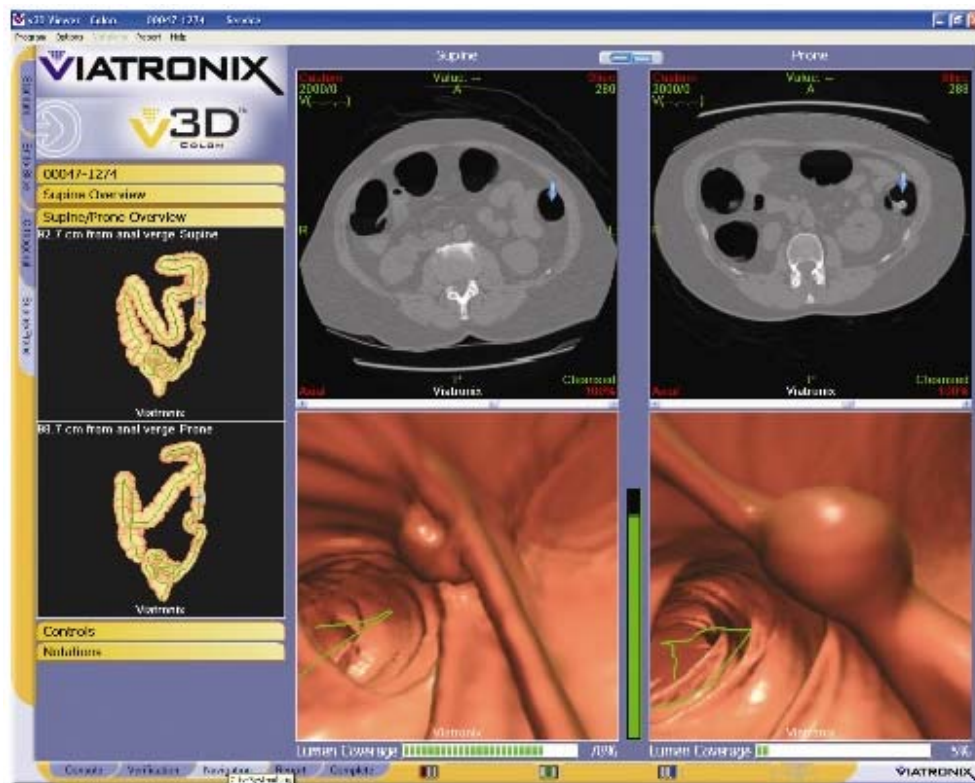


Figura 3.2: Interfaz del sistema *V3D* de *Viatronix*.

- *VoxelView / Vitrea2 de Vital Images*

A mediados de los noventa Vital Images desarrolló el sistema *VoxelView*. Se trataba de una aplicación que utilizaba *Direct Volume Rendering* para visualizar el volumen de datos. Incorporaba un sistema de navegación automático basado en la definición de *key frames* por el usuario y en la interpolación del resto de la secuencia. A pesar de la gran calidad de las imágenes obtenidas, el principal inconveniente de este sistema era el tiempo que tardaba en generar las animaciones y la poca flexibilidad en su visionado (tan solo play/replay de la animación).

Actualmente Vital Images ofrece al mercado el sistema *Vitrea 2*, basado en texturas 2D y 3D para implementar la visualización mediante *DVR*. Incorpora un sistema de navegación manual no interactivo que permite al usuario recorrer el volumen paso a paso.

3.4 Conclusiones

En las tablas 3.1 y 3.2 se muestra de manera resumida cuales son las características principales de los sistemas analizados en este capítulo. Se comparan las técnicas utilizadas para la captación de los datos, la segmentación, la reconstrucción de isosuperficies, la visualización y la inspección del volumen.

Sistema	Input			Segmentación					Reconstrucción
	CT	MRI	Otros	Thresholding	Boundary localization	Region Growing	Otros	No utiliza	Marching Cubes
<i>FreeFlight</i>	✓	✓		✓		✓			✓
<i>VICON</i>	✓					✓			✓
<i>VIVENDI</i>	✓	✓	✓			✓			✓
<i>VirEn</i>	✓	✓				✓			✓
<i>CRS4</i>	✓	✓						✓	
<i>3D Slicer</i>	✓	✓		✓	✓		✓		✓
<i>VESA</i>	✓	✓		✓	✓		✓		✓
<i>V3D Viewer</i>	✓	✓	✓			✓	✓		
<i>VoxelView (*)</i>	?	?	?	?	?	?	?	?	?
<i>Vitrea2 (*)</i>	?	?	?	?	?	?	?	?	?

(*) Se trata de visualizadores de volumen a los que hay que pasar los datos segmentados o clasificados.

Tabla 3.1: Métodos utilizados en la captación y segmentación de los sistemas descritos.

Sistema	Visualización - Rendering			Sistema de navegación		
	Superficies poligonales	Direct Volume Rendering		Automático	Manual	Guiado
		Ray-casting	Texturas 3D			
<i>FreeFlight</i>	✓			✓	✓	
<i>VICON</i>	✓	✓		✓		✓
<i>VIVENDI</i>	✓					✓
<i>VirEn</i>	✓	✓		✓		✓
<i>CRS4</i>			✓		✓	
<i>3D Slicer</i>	✓			✓	✓	
<i>VESA</i>	✓			✓	✓	
<i>V3D Viewer</i>	✓	✓				✓
<i>VoxelView</i>		✓		✓		
<i>Vitrea2</i>		✓			✓	

Tabla 3.2: Métodos utilizados en la visualización y navegación de los sistemas descritos.

En la tabla 3.1 puede observarse que los sistemas de endoscopia virtual analizados obtienen la información mayoritariamente a partir de *CT* y *MRI*. Si la técnica de visualización se basa en superficies poligonales han de segmentar el volumen y reconstruir las isosuperficies, para lo que se utiliza la técnica de *Marching Cubes* [14]. Los sistemas que ofrecen únicamente *Direct Volume Rendering*, como es el caso de *VoxelView* y *Vitrea2*, no han de realizar la etapa de segmentación, sino que realizan directamente la clasificación de los datos del volumen.

Como muestra la tabla 3.2, la mayoría de sistemas utilizan la visualización de isosuperficies. Esto se debe a que la visualización de mallas poligonales permite obtener un *frame rate* elevado durante la inspección del modelo, en cambio la calidad de las imágenes obtenidas es inferior al de las otras técnicas. Por este motivo algunos sistemas implementan *ray-casting*, así pueden obtener vistas concretas de mayor calidad.

En cuanto a las técnicas de exploración utilizadas, la mayoría de sistemas incorporan el modelo de navegación automática acompañado de otras modalidades que permiten al usuario navegar con mayor libertad. Para los sistemas que ofrecen el modelo de navegación manual, la detección de colisiones suele estar sujeta a un conjunto de restricciones físicas que limitan el movimiento de la cámara virtual [5] [7] [9].

La principal limitación de estos sistemas es que para conseguir visualizaciones de calidad, el tiempo de cálculo que se consume es muy elevado. Tan solo el sistema *CRS4* utiliza las texturas 3D como método de visualización, debido a que resulta complicado obtener inspecciones interactivas sin el hardware gráfico más reciente. Los esfuerzos de los investigadores se centran en mejorar la relación calidad – interactividad de las endoscopias virtuales, así como en la investigación de nuevas técnicas que permitan disminuir el tiempo de cálculo de las etapas que intervienen en el proceso: segmentación, generación automática de trayectorias para la inspección, etc.

En el siguiente capítulo se analizan cuales son los factores que influyen en la calidad del uso de texturas 3D cuando se aplican a la visualización de volumen en general, y en la simulación de endoscopias en estructuras tubulares en particular. También se presentan las soluciones implementadas para mejorar la relación calidad – interactividad, y se evalúan los resultados obtenidos.

Capítulo 4

Visualización de endoscopias virtuales con texturas 3D

4.1 Introducción

En capítulos anteriores se han presentado las etapas básicas en la visualización de datos de volumen, las etapas específicas para realizar endoscopias virtuales y los sistemas desarrollados con este fin. Se ha comentado que el objetivo que se persigue en estos sistemas es el de obtener la relación óptima entre calidad e interactividad.

Diversos aspectos influyen en la calidad de la visualización final, desde la técnica utilizada hasta la precisión con la que se desarrolla cada etapa del proceso. Cuanto mayor es la calidad que se desea conseguir, mayor es el tiempo de cálculo invertido. En este capítulo y en primer lugar, se describen los aspectos que influyen en la calidad de la visualización con texturas 3D para realizar endoscopias virtuales. Posteriormente, se analizan las soluciones propuestas para reducir los *artifacts* resultantes y se presentan las técnicas implementadas para mejorar la relación calidad – interactividad. Por último, se evalúan los resultados obtenidos con el sistema implementado, cuya descripción se presenta en el anexo A.

4.2 Proceso de visualización con texturas 3D

El *pipeline* gráfico que rige la visualización con texturas 3D consta de los siguientes pasos:

- *Etapa de muestreo (sampling)*: Muestreo del volumen (textura 3D) mediante planos de corte para obtener el valor de los *voxels*.
- *Etapa de filtrado (filtering)*: interpolación del valor de los *voxels* para reconstruir el volumen, es decir, para obtener el valor de propiedad en un punto cualquiera del volumen.

- *Etapa de clasificación (classification)*: asignación de las propiedades ópticas a los valores del volumen.
- *Cálculo de la iluminación (shading)*: cálculo del color. Pueden tenerse en consideración fuentes de luz externas para calcular la iluminación del volumen.
- *Etapa de composición (blending)*: composición del color de las muestras que se proyectan en un mismo píxel.

En cada una de estas etapas se realizan diferentes aproximaciones que pueden influir en la calidad de la visualización final (ver figura 4.1). A continuación se describen los principales aspectos a considerar en cada paso y algunas soluciones que han sido propuestas para disminuir o eliminar sus efectos.

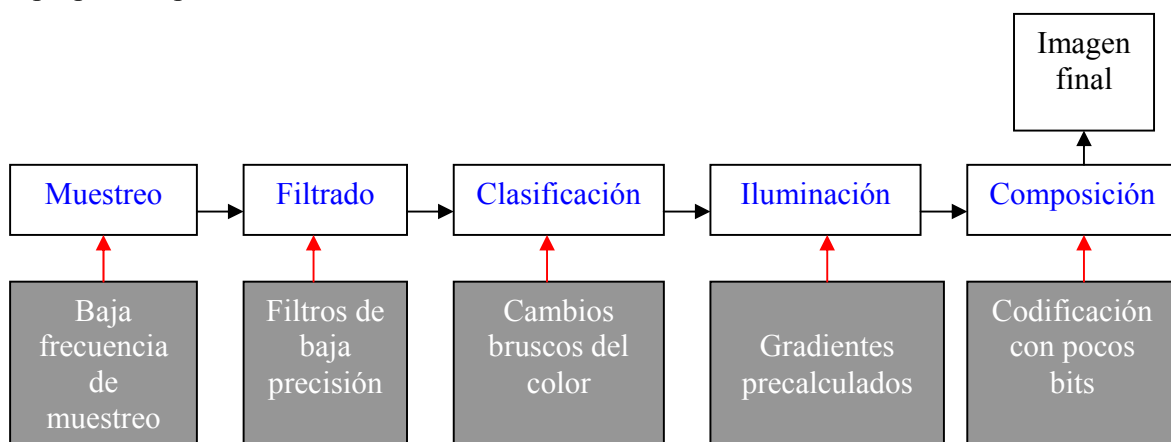


Figura 4.1: Pipeline gráfico en la visualización con texturas 3D. En cada etapa intervienen factores que pueden generar artifacts.

• *Etapa de muestreo*

Uno de los factores principales que influyen en la calidad de las imágenes es la frecuencia de muestreo del volumen. A menores frecuencias, mayor distancia entre las muestras y menor calidad en el resultado final (se pierde información). Para reconstruir el volumen a partir del conjunto de muestras sin perder información, se ha de garantizar que cada *voxel* queda muestreado por algún plano. Enviando a pintar el número de planos que se obtiene de la ecuación 4.1, se asegura esta premisa en la diagonal mayor del volumen, lo que garantiza como mínimo 1 plano por *voxel* sea cual sea la dirección de visión.

$$\text{Núm. planos} = \sqrt{\text{ancho del volumen}^2 + \text{alto del volumen}^2 + \text{profundidad volumen}^2}$$

Ecuación 4.1: Cálculo del número de planos para tener 1 muestra por voxel en la diagonal mayor del volumen.

Cuando la distancia entre muestras es elevada, se puede apreciar en las imágenes un conjunto de imperfecciones conocidas como *wood-grainy artifacts* (ver figura 4.2). La solución para eliminar estos errores consiste en disminuir la distancia entre muestras aumentando el número de planos que se envían a pintar. Cuando se realizan visualizaciones con proyección perspectiva (como es el caso de las endoscopias virtuales), la distancia entre las muestras de planos consecutivos aumentan en función del ángulo de obertura de la cámara, lo que provoca que para grandes ángulos se magnifiquen este tipo de *artifacts* (ver apartado 4.4).

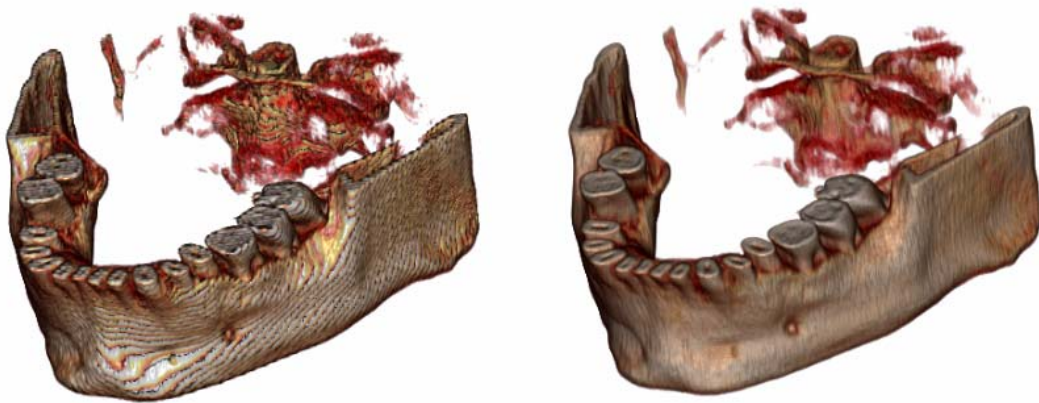


Figura 4.2: Wood-grainy artifacts debidos a la baja frecuencia de muestreo (izq.) y resultado de aumentar el número de muestras (der.).

- *Etapa de filtrado*

La etapa de filtrado es la que permite reconstruir el modelo de volumen a partir de las muestras, o dicho en otras palabras, obtener el valor de propiedad de los puntos del volumen que no están muestreados. Esto se consigue aplicando un filtro de reconstrucción al conjunto de muestras.

Existen varios tipos de filtros que se pueden utilizar en esta etapa [3], con los que se consiguen reconstrucciones de mayor o menor calidad. Se ha de tener en cuenta que los filtros más precisos requieren un tiempo de cálculo mayor, por lo que su aplicación puede penalizar notablemente al rendimiento del proceso. El filtro más utilizado con texturas 3D es la interpolación trilineal, ya que la efectúa directamente OpenGL a través del hardware.

- *Etapa de clasificación*

Como se ha comentado en el capítulo 2, en la fase de clasificación se utilizan funciones de transferencia para asignar un color y una opacidad a cada muestra. El orden en el que se realiza el filtrado y la clasificación de los datos afecta a la calidad de las imágenes resultantes, como se puede observar en la figura 4.3.

Si la clasificación se realiza antes que el filtrado (primero se asigna un color a cada *voxel*), las propiedades ópticas de las muestras del volumen se interpolan para calcular las correspondientes a un punto concreto en la etapa de filtrado. Si por el contrario se efectúa primero el filtrado, se interpolan los valores de propiedad de las muestras para calcular el valor de propiedad en un punto concreto, y después se le asignan un color y una opacidad.

Cuando las funciones de transferencia que se le aplican al volumen no son lo suficientemente precisas, es posible que se puedan apreciar cambios bruscos de color cuando la etapa de filtrado precede a la de clasificación. Para evitar que se vean estos *artifacts* se puede reducir la distancia de muestreo, pero como se analiza en el apartado 4.3, el principal inconveniente es que se reduce notablemente el *frame rate* de la visualización.

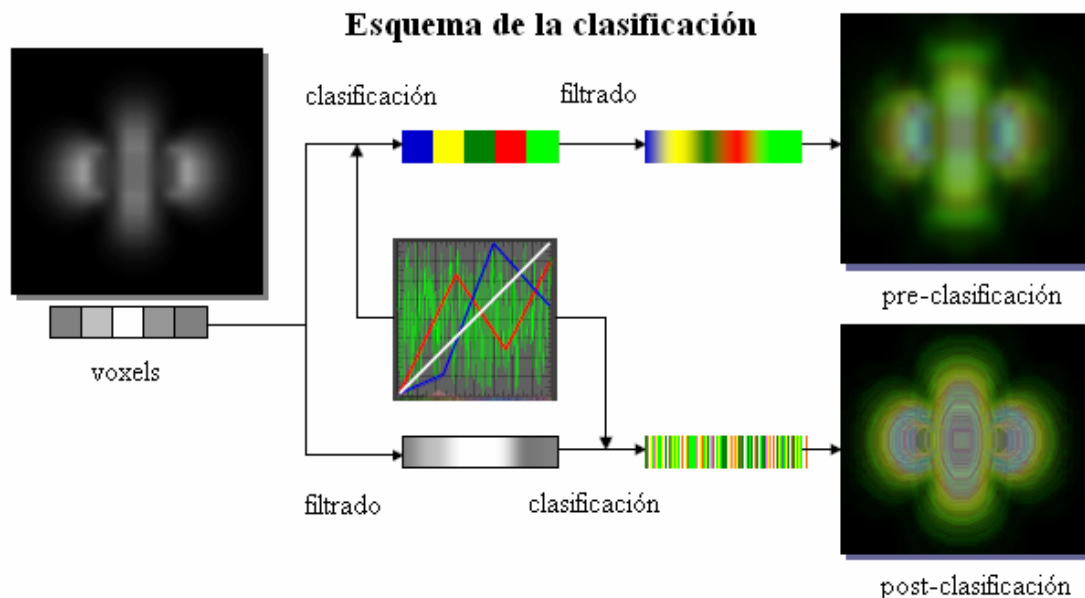


Figura 4.3: Comparación del proceso y de los resultados obtenidos variando el orden del filtrado y la clasificación. Figura extraída de [3].

- *Cálculo de la iluminación*

Para realizar el cálculo del color en un punto de acuerdo con las condiciones de iluminación (*shading*), es necesario calcular un gradiente por *voxel* (o en cada punto) a partir de la información de los *voxels* vecinos. Estos cálculos se pueden hacer durante un preproceso o bien en tiempo de ejecución, según las prestaciones del hardware gráfico utilizado.

Cuando los cálculos se realizan en un preproceso, se calculan los gradientes y se guardan normalizados en los 8 bits de los canales RGB de una textura de volumen. En este caso, para no afectar al rendimiento, los valores del volumen se suelen almacenar en el canal *alpha* (A) de la textura de gradientes, permitiendo obtener estos datos con un simple acceso a textura.

El hecho de guardar los componentes del gradiente en tan solo 8 bits puede generar algunos *artifacts*. Además, durante el proceso de filtrado se interpolan los gradientes y el resultado no tiene porqué estar normalizado. Esto puede ser otra causa de errores cuando el hardware gráfico no permite la normalización de gradientes por fragmento en tiempo de ejecución. Una posible solución es almacenar los gradientes en texturas 3D de mayor precisión (16 o 32 bits por canal).

Cuando el *shading* se realiza en tiempo de ejecución, el gradiente se calcula en cada fragmento a partir del valor de propiedad de los *voxels* vecinos al que se proyecta en el fragmento. En relación a la calidad que se pretenda conseguir, se utiliza la información de más o menos vecinos, 6 en caso de calcular el gradiente con la técnica *central-differences*, 26 si se calcula aplicando el procedimiento de *Sobel*, pero siempre teniendo en cuenta que cuantos más accesos a textura se realizan, menor es el rendimiento del proceso y menores *frame rates* se consiguen.

- *Etapas de composición*

El último paso en el *pipeline* gráfico consiste en realizar la composición del color a partir de los colores cada muestra. Debido a que los colores en esta etapa suelen estar codificados en 8 bits, la acumulación de errores durante la composición tiende a crecer rápidamente. Este hecho puede afectar a la calidad de la imagen final en forma de algunas imperfecciones en el color de ciertas zonas del volumen. Una de las soluciones propuestas en la literatura es aumentar el número de bits con los que se realiza el *blending*.

4.3 Análisis de las soluciones planteadas

Para evaluar el rendimiento y los resultados que se pueden obtener con la visualización mediante texturas, se ha implementado un sistema que permite realizar endoscopias virtuales (ver anexo A). Esta aplicación se ha utilizado para analizar la calidad de las imágenes, según la técnica aplicada en cada etapa del *pipeline* gráfico (ver figura 4.1).

Se ha comentado que para eliminar los *artifacts* producidos por la frecuencia de muestreo hay que disminuir la distancia entre muestras aumentando el número de planos (ver apartado 4.2). No obstante, hay que tener en cuenta que el número de planos influye directamente en la velocidad de interacción con el modelo, de manera que a mayor número de planos mayor es el tiempo que se tarda en generar cada imagen.

Una posible solución para mantener la interactividad y obtener imágenes de calidad es utilizar la técnica conocida como *adaptive sampling*. Esta técnica se basa en reducir la distancia de muestreo en aquellas zonas del volumen con mayor variabilidad, mientras que se reduce (o se mantiene a 1 plano por *voxel*) en las zonas más homogéneas; no obstante, la aplicación de esta técnica comporta un conocimiento a priori del volumen a analizar. En el apartado 4.4 se discuten diferentes alternativas para modificar la frecuencia de muestreo y su efecto en la calidad e interactividad de la aplicación.

Respecto a los *artifacts* producidos en las etapas de filtrado y clasificación, se ha observado que cuando la resolución del volumen es elevada y se desea (como en el caso de las endoscopias virtuales) resaltar la superficie de las estructuras, efectuar el filtrado antes que la clasificación conduce a visualizaciones de calidad (similares a las obtenidas con *ray-casting*).

Cuando en la etapa de *shading*, el cálculo de la iluminación (cómputo del gradiente – normal en el punto – aplicación del modelo empírico de *Phong/Lambert*) se lleva a cabo en tiempo de ejecución mediante un *shader*, no se observan los defectos visuales que se pueden originar en esta etapa.

Para evaluar la influencia en la visualización final del *blending*, se ha utilizado un *frame buffer object* (FBO) que permite codificar los colores con más bits que los 8 que utiliza el *frame buffer* clásico de OpenGL en la composición. En los resultados obtenidos de las pruebas realizadas, no se han apreciado diferencias visuales cuando se utilizan 8 bits en el *blending* respecto a cuando se utilizan 16 bits, en cambio si que se han podido observar cuando se han utilizado 2 y 4 bits.

En la imagen de la izquierda de la figura 4.4 (2 bits para la composición), se pueden observar un conjunto de puntos claros distribuidos uniformemente, mientras que en la imagen de la derecha (8 bits) desaparecen totalmente. Para no perjudicar al rendimiento de la aplicación, se considera que es razonable mantener los 8 bits en este proceso.

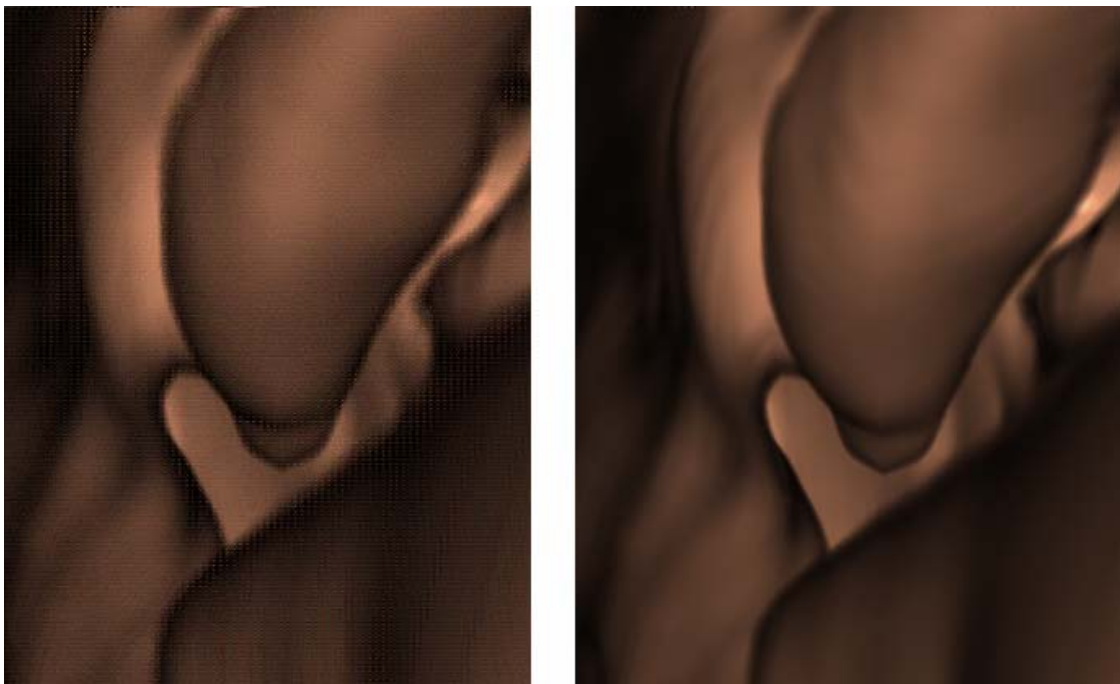


Figura 4.4: Imágenes obtenidas con una precisión de 2 bits (izq.) y 8 bits (der.) durante la etapa de blending.

4.4 Técnicas de optimización para la inspección interactiva

Tras el análisis de los errores que se introducen en el proceso de visualización con texturas 3D (apartado 4.3), hemos concluido que para alcanzar una buena relación calidad – interactividad se debe escoger una técnica adecuada de *adaptive sampling*. Se han implementado diversas modalidades que se presentan en este apartado y cuyos resultados experimentales se evalúan en el apartado 4.5. Por otra parte, también se ha propuesto un sistema de detección de colisiones simple y eficiente que no afecta al rendimiento general de la aplicación y que permite la navegación interactiva por el interior de estructuras tubulares sin salirse de ellas.

4.4.1 Adaptive sampling

En el apartado 4.2 se ha comentado que para eliminar los *artifacts* producidos por tener bajas frecuencias de muestreo, se debe reducir la distancia entre muestras aumentando el número de planos con los que se visualiza el volumen. También se ha concluido que estos tipos de *artifact* se ven acentuados en el caso de utilizar una cámara perspectiva durante la navegación.

El principal inconveniente de aumentar el número de planos es que aumenta el tiempo de la visualización dado que se han de pintar más polígonos. Las técnicas de *adaptive sampling* permiten mantener un número de planos constante y razonable (a efectos de la interactividad). Para ello se reduce la distancia de muestreo en las zonas del volumen donde se quiere obtener mayor precisión visual mientras que se incrementa o mantiene en las otras.

Una primera consideración a tener presente cuando se utiliza *adaptive sampling* es que la distancia entre muestras (planos) al componer la opacidad no es constante, a diferencia de lo que se considera habitualmente. Si no se tiene en cuenta este dato, al componer el color de la forma habitual se obtienen regiones más brillantes en las zonas con menor distancia entre muestras (ver figura 4.5).

La ecuación 4.2 muestra una expresión válida [3] para realizar la corrección de opacidad. El nuevo cálculo depende tanto de la opacidad acumulada (α_{acum}) como de la distancias de muestreo actual y base ($\Delta x / \Delta x_0$). Aunque esta modificación en el cálculo de la opacidad es precisa para proyecciones axonométricas, continúa siendo aproximada para proyecciones perspectivas, ya que la distancia entre las muestras de dos planos consecutivos no es constante para todos los píxeles de la imagen (ver figura 2.12).

$$\alpha \text{ corregida} = 1 - [1 - \alpha_{\text{acum}}]^{\Delta x / \Delta x_0}$$

Ecuación 4.2: Cálculo de la opacidad con adaptive sampling.

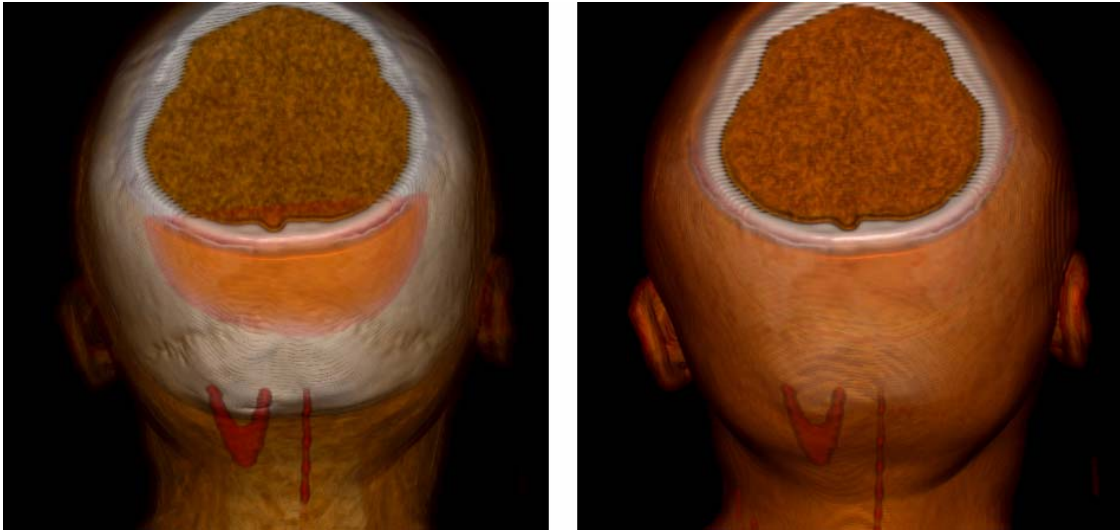


Figura 4.5: Resultado de aplicar (der.) o no (izq.) la corrección de opacidad.

A continuación se describen las diferentes modalidades de *adaptive sampling* evaluadas. Todas ellas se basan en la hipótesis de que el sistema de visualización utiliza una única textura 3D con la información del volumen y mantiene siempre los planos de muestreo perpendiculares a la dirección de visión (esta hipótesis es la más usual). Por otra parte, el plano de recortado anterior (Z_{near}) de la cámara coincide con el plano más cercano al observador en caso de que éste se encuentre situado fuera del volumen, mientras que se coloca justo delante del observador si se navega por el interior (como es el caso que nos ocupa). El plano de recortado posterior (Z_{far}) se hace coincidir con el plano que muestrea los *voxels* más alejados del observador.

Como se ha explicado en el apartado 2.6.2, para inspecciones de estructuras tubulares es deseable disponer de la proyección perspectiva, ya que proporciona mayor sensación de profundidad. En el apartado 4.2 se ha comentado que la deformación perspectiva magnifica los *artifacts* debidos a la frecuencia de muestreo, esto se debe principalmente a que la distancia entre las muestras que se proyectan en píxeles adyacentes no es constante. Si esta distancia fuese similar, aunque no la misma, los errores producidos al calcular el color serían imperceptibles. Para grandes ángulos de obertura de la cámara, las distancias entre las muestras que se proyectan en píxeles adyacentes crecen, por lo que también aumentan los *artifacts*. En la figura 4.6 se pueden ver tres vistas en perspectiva de las fosas nasales con diferentes ángulos de obertura; obsérvese en las paredes de la cavidad como los errores debidos a la frecuencia de muestreo son cada vez más significativos.

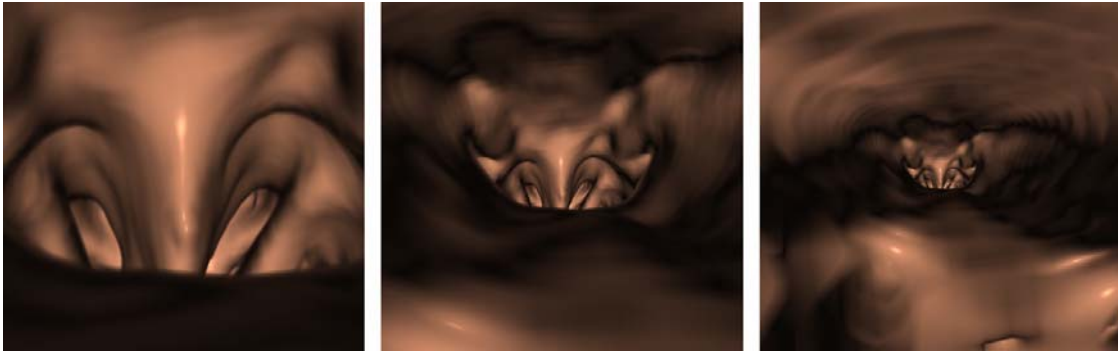


Figura 4.6: Vista perspectiva de las fosas nasales con ángulos de abertura de 30° (izq.), 80° (centro) y 130° (der.).

Se han evaluado 4 modalidades de *adaptive sampling* que varían en la forma de distribuir los planos entre Z_{near} y Z_{far} , de manera que se disminuye la distancia entre los planos más cercanos al observador y se aumenta entre los planos más alejados. Las cuatro técnicas estudiadas son las siguientes:

- *Modalidad 1*: agrupar hasta una cierta distancia respecto al observador $2/3$ de los planos. Esta distancia corresponde con $1/4$ de la distancia entre Z_{near} y Z_{far} .
- *Modalidad 2*: agrupar hasta una distancia fija respecto al observador $2/3$ de los planos. Esta distancia fija corresponde con $1/4$ de la diagonal del volumen.
- *Modalidad 3*: calcular los planos a pintar de forma que siempre se tengan 2 muestras por *voxel*.
- *Modalidad 4*: calcular los planos a pintar de forma que haya 2 muestras por *voxel* hasta una cierta distancia del observador ($1/4$ de la diagonal del volumen) y 1 muestra en el resto.

Los resultados experimentales obtenidos para cada una de las modalidades implementadas demuestran que con todas ellas se consigue mayor calidad que distribuyendo los planos uniformemente a lo largo del volumen. En el apartado 4.5 se realiza un análisis detallado de los resultados, pero como síntesis se puede concluir que en la tercera y la cuarta modalidad no se eliminan del todo los *artifacts*, a pesar de que se reducen notablemente. Con la primera variante se pueden apreciar algunos defectos visuales en las zonas con poca frecuencia de muestreo ($1/3$ de los planos en la parte del volumen más alejada del observador), mientras que con la segunda modalidad se consiguen eliminar por completo. En la figura 4.7 se puede ver como quedan distribuidos los planos para cada una de las modalidades anteriores.



Figura 4.7: Distribución de los planos en las variantes de adaptive sampling implementadas. Modalidad 1 (arriba, izquierda), modalidad 2 (arriba, derecha), modalidad 3 (abajo, izquierda) y modalidad 4 (abajo, derecha).

4.4.2 Adaptación del viewport

Independientemente de la modalidad de *adaptive sampling* empleada, cuando se tiene que mantener un elevado número de planos (grandes distancias entre Z_{near} y Z_{far}) se reduce el grado de interacción. A efectos de conseguir un *frame rate* aceptable durante la navegación, se ha analizado el efecto de realizar la visualización en un *viewport* de menor precisión y ampliar la imagen obtenida. De este modo se reduce tanto el tiempo de *rasterización* como el número de fragmentos resultantes y, por tanto, el número de los cálculos que intervienen en el *shading* y la composición del color (los cálculos realizados en el *fragment shader*).

La idea básica de la técnica de aceleración es visualizar el volumen en un *viewport* de tamaño reducido, almacenar la imagen en una textura 2D y posteriormente, mapear la textura en un plano del tamaño del *viewport* de la aplicación.

Se han realizado pruebas con *viewports* de diferentes tamaños observando que la calidad de las imágenes finales se mantiene, siempre que el tamaño del *viewport* reducido sea como mínimo la mitad del de la aplicación. Para tamaños inferiores, los filtros lineales de amplificación (un *texel* se mapea en varios píxeles) de texturas 2D en OpenGL no consiguen obtener resultados lo suficientemente precisos y se pueden observar algunos errores de *aliasing* en las imágenes finales. Las figuras 4.8 y 4.9 muestran estos *artifacts* en la visualización y durante la inspección de un modelo cuando el tamaño del *viewport* reducido es de 128 x 128 píxeles y 64 x 64 píxeles respectivamente.

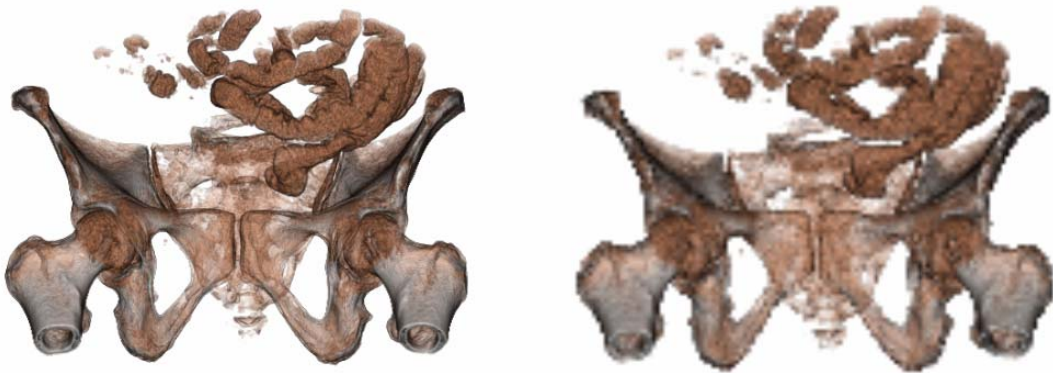


Figura 4.8: Resultados obtenidos con el *viewport* original de 464 x 448 píxeles (izq.) y uno reducido de 128 x 128 píxeles (der.).

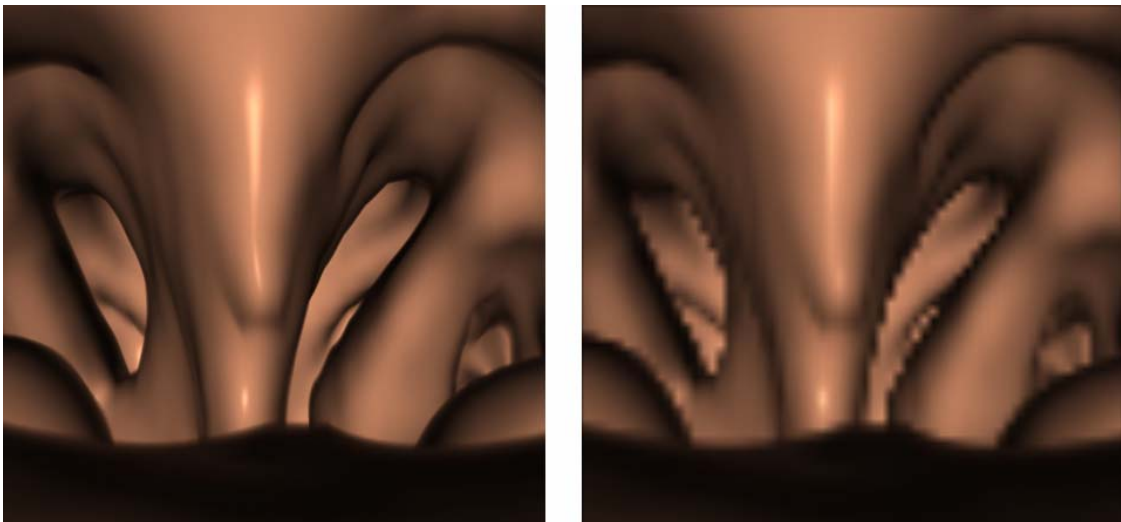


Figura 4.9: Resultados obtenidos con el *viewport* original de 464 x 448 píxeles (izq.) y uno reducido de 64 x 64 píxeles (der.).

Como se discute en el apartado 4.5, con esta técnica de aceleración se consiguen *frame rates* más elevados en la inspección de los modelos de volumen, obteniendo imágenes con la misma calidad que las originales.

4.4.3 Detección de colisiones

En una endoscopia virtual se requiere implementar un método de navegación que permita al especialista inspeccionar las estructuras anatómicas de manera amigable y sin salirse de ellas. Una de las limitaciones de los sistemas existentes de endoscopia virtual (ver apartado 3.4) es que el tiempo que se consume para realizar los cálculos automáticos de generación de trayectorias es elevado [7], limitando su aplicabilidad clínica.

Nuestra propuesta es muy simple, consiste en trazar un rayo desde el observador (posición actual) en la dirección de movimiento, calcular la primera intersección del rayo con las paredes del conducto, y permitir el avance de la cámara sólo hasta una distancia de seguridad (preestablecida y modulable) del punto de intersección (ver figura 4.9).

Para saber el punto de intersección, se consulta el valor de la propiedad en diferentes puntos del rayo separados por un cierto incremento (modulable según la precisión que se desee obtener), hasta encontrar el punto con el mismo valor que la superficie. Dependiendo del incremento entre los puntos del rayo, hay ocasiones en las que no se encuentra el valor de la superficie, sino que para dos puntos consecutivos A y B del rayo, uno tiene un valor inferior y el otro superior. En este caso se puede obtener por interpolación lineal el punto entre A y B en el que se encuentra la intersección (con el valor de la superficie).

Para acelerar los cálculos en la detección de colisiones, se ha implementado una pequeña variante del algoritmo expuesto. En lugar de recorrer el rayo hasta encontrar la primera intersección, se traza un rayo de longitud limitada. Si no se encuentra la intersección en esa longitud se permite el avance hasta la posición de destino.

```
Rayo r = crear_rayo (observador, direccionMovimiento);
Punto3D interseccion = calcular_interseccion (rayo, modelo, valorSup);
int dist = distancia (observador, interseccion);

if (distancia >= margenDeSeguridad){
    actualizar_posicion (direccionMovimiento);
}
```

Figura 4.9: Pseudo-código del algoritmo de detección de colisiones.

Esta técnica para detectar colisiones es sencilla, fácil de implementar y no disminuye el rendimiento de la visualización. La precisión en la detección de las colisiones está condicionada principalmente por el incremento aplicado al recorrer el rayo y por la exactitud de la clasificación teniendo en cuenta el valor de la propiedad. Normalmente habrá que definir un rango de valores en vez de utilizar un valor exacto, para evitar las imprecisiones de la clasificación.

Para realizar endoscopias virtuales, se han implementado los tres sistemas de navegación utilizados hasta la fecha: navegación manual, navegación automática y navegación guiada. A continuación se describe con detalle cada uno de estos modelos.

- *Navegación manual*

Este modelo de navegación permite al usuario manejar una cámara virtual por el interior del conducto, sin estar sujeto a condiciones físicas que puedan restringir el recorrido durante la exploración, aparte de la colisión con las superficies visualizadas. De esta manera el especialista puede inspeccionar las estructuras anatómicas desde diferentes ángulos y posiciones sin la necesidad de seguir un camino precalculado.

- *Navegación automática*

El modelo de navegación automática se basa en la definición de unos puntos clave (*key frames*) por parte del usuario, a partir de los cuales la aplicación genera la animación. Para ello se registra la posición del observador, la dirección de visión y la orientación de la cámara en cada *key frame* y se calculan estos valores por interpolación en los *frames* intermedios.

Mientras que la posición del observador se va actualizando en cada *frame*, la orientación de la cámara y la dirección de visión tan solo se actualizan para posiciones cercanas a los puntos predefinidos por el usuario. De esta manera se consiguen transiciones suaves y se mantienen las direcciones de visión originales durante la mayor parte de la animación.

- *Navegación guiada*

El modelo de navegación guiada se basa en recorrer el camino seguido en la navegación automática de forma que el especialista controla la dirección de visión de la cámara. A diferencia de otros sistemas, la técnica implementada permite cambiar a navegación manual para inspeccionar con detalle las diferentes zonas que se visualizan, con la posibilidad de volver a la trayectoria seguida por el camino de forma automática.

4.5 Discusión y evaluación de resultados

A continuación se presentan y discuten los resultados obtenidos con las técnicas propuestas en el apartado 4.4, además de comparar los resultados con los métodos de visualización de isosuperficies y *ray-casting*. Para realizar esta evaluación se ha desarrollado una aplicación que permite visualizar un modelo de volumen mediante texturas 3D, *ray-casting* e isosuperficies (ver anexo A). Como base para la implementación, se ha partido del software existente en el grupo de investigación MOVING, que se ha tenido que adaptar a la visualización perspectiva. También se han incorporado las cuatro modalidades de *adaptative sampling*, los tres sistemas de navegación por estructuras tubulares y el algoritmo de detección de colisiones.

Para realizar las diferentes pruebas que aquí se presentan, no se ha precisado utilizar ningún sistema hardware específico, sino que se han llevado a cabo en un PC convencional con las características descritas en la tabla 4.1.

Equipo utilizado	
<i>Procesador</i>	Intel Core 2 Duo E6300, 1866 MHz
<i>Tarjeta gráfica</i>	Radeon X1950 Pro (256 MB)
<i>Acelerador 3D</i>	ATI Radeon X1950 Pro (RV570)

Tabla 4.1: Características del hardware utilizado.

En las pruebas realizadas se han utilizado dos modelos de volumen de diferente tamaño y resolución, obtenidos a partir de imágenes *CT*. El primero es un modelo de la cabeza de 256 x 256 x 196 *voxels* en el que se ha simulado la navegación por la parte alta del esófago y las fosas nasales del paciente, mientras que el segundo es un modelo del torso y el abdomen de 512 x 512 x 124 *voxels*, en el que se inspecciona el interior de la aorta a la altura del corazón. Las características principales de estos modelos se detallan en la tabla 4.2.

Volumen	Cabeza	Torso – Abdomen
<i>Entrada</i>	CT	CT
<i>Imágenes</i>	196	124
<i>Resolución de imagen</i>	256 x 256	512 x 512
<i>Núm. voxels</i>	256 x 256 x 196	512 x 512 x 124
<i>Voxel Size X (Δx)</i>	1.023	0.626
<i>Voxel Size Y (Δy)</i>	1.023	0.626
<i>Voxel Size Z (Δz)</i>	1.0	2.0
<i>Dimensiones volumen</i>	262.0 x 262.0 x 196.0	320.99 x 320.99 x 248.0

Tabla 4.2: Características de los modelos de volumen.

El primer conjunto de tests realizado ha tenido como objetivo comparar la relación calidad – interactividad conseguida con las diferentes técnicas de visualización analizadas en el capítulo 2: visualización de isosuperficies, *ray-casting* y texturas 3D. Para evaluar los tiempos se han generado diversas animaciones y se presentan los *frame rates* mínimos y máximos obtenidos.

En la tabla 4.3 se muestra el tamaño de las mallas de triángulos generadas para cada uno de los volúmenes en la visualización de isosuperficies. La segmentación del volumen se ha llevado a cabo con la técnica de *thresholding* (ver apartado 2.4) estableciendo los límites de las superficies a segmentar (valores de propiedad entre 100 y 4000 para la cabeza y 1300 y 1600 en el caso del abdomen). Para reconstruir las isosuperficies se ha utilizado el algoritmo de *Marching Cubes*. Como el modelo del abdomen es más grande y de mayor resolución que el de la cabeza, el tiempo que se emplea en la segmentación y generación de la malla es superior. Por el mismo motivo, el número de triángulos de la malla del abdomen también es superior. A pesar de la diferencia de tamaño, para ambos casos se consiguen *frame rates* interactivos (superiores a 25 fps) con el equipo utilizado.

	Esófago	Aorta
Núm. Triángulos	412 988	1 425 032
Núm. Vértices	206 590	720 780
Tipo segmentación	thresholding	thresholding
Tiempo generar malla	5 – 6 segundos	14 – 15 segundos
Frame rate	27 fps	27 fps

Tabla 4.3: Resultados de la visualización de isosuperficies.

En cuanto a los resultados visuales, la figura 4.10 muestra dos vistas concretas de cada uno de los modelos, con la calidad suficiente como para permitir la realización de endoscopias virtuales con esta técnica de visualización.

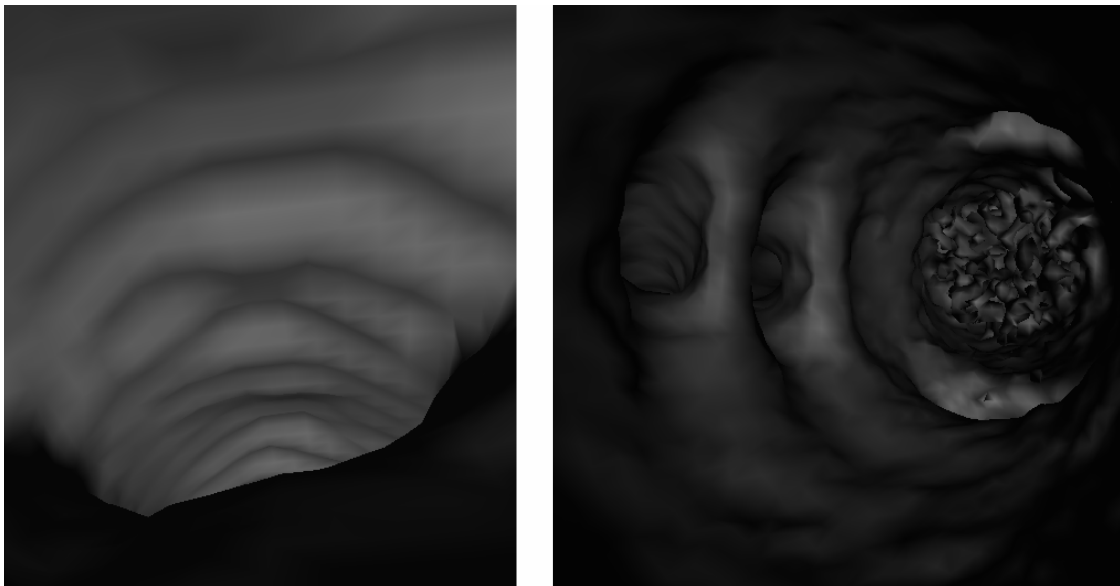


Figura 4.10: Imágenes obtenidas con la visualización de isosuperficies del esófago (izq.) y de la aorta (der.).

Se ha observado que al visualizar la superficie desde puntos cercanos (grandes factores de *zoom*), pueden surgir algunos defectos visuales. Estas imperfecciones son conocidas en la literatura como *diamond artifacts* y son consecuencia de tener vectores normales muy diferentes en los vértices de una misma cara. La forma de diamante es debida a que con *Marching Cubes* convergen en cada vértice 4 aristas. Si se aplica el modelo de Phong al realizar el *shading*, aunque no se acaban de eliminar este tipo de *artifacts* se disimulan en gran medida (ver figura 4.11). Este tipo de *shading* tan solo se puede utilizar de forma interactiva si se implementa con la tarjeta gráfica.

El principal inconveniente de estas visualizaciones es que la precisión del resultado depende en gran medida de la segmentación y solo permite observar las estructuras del volumen que se han tenido en cuenta durante la segmentación (si se ha segmentado el esófago o la aorta, no permite ver las demás estructuras del modelo).

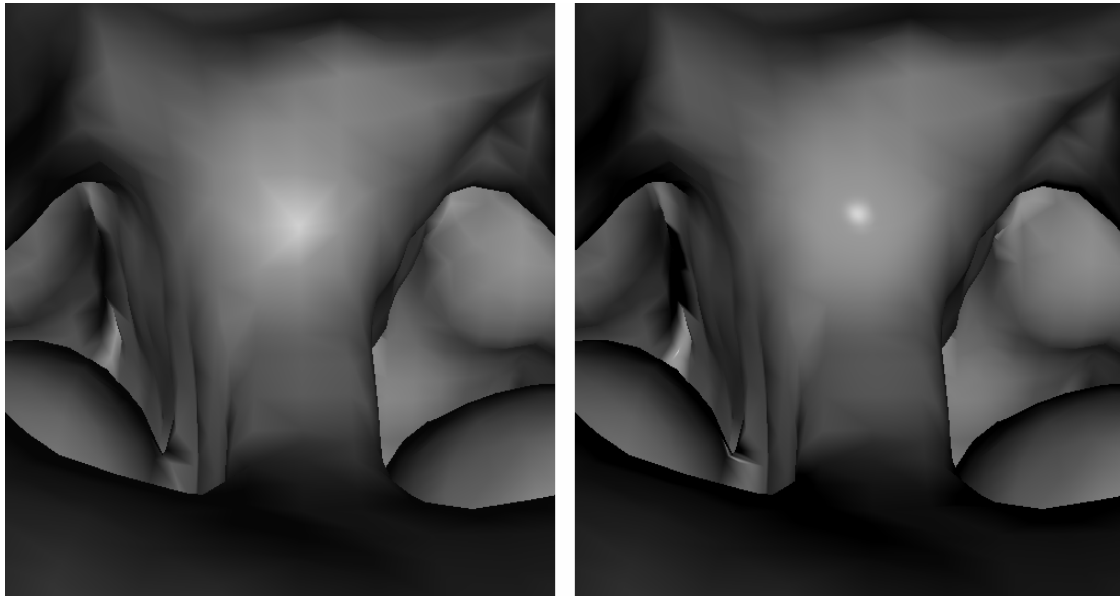


Figura 4.11: Diamond artifacts visibles con el modelo de Gouraud (izq.) y con el modelo de Phong (der.)

Utilizando la técnica de *ray-casting* se obtienen imágenes de mayor calidad que con la visualización de isosuperficies, sobretodo si se realiza un muestreo suficientemente preciso (ver figuras 4.12 y 4.13). A pesar de ello, en la tabla 4.4 se puede ver que para un *viewport* de tamaño 464 x 448 píxeles, el tiempo que se tarda en generar cada imagen es de varios segundos, y superior al minuto con 2 muestras por *voxel* en el abdomen. Aunque no se han utilizado técnicas específicas de aceleración de cálculos (como la detección de isosuperficies para concluir la composición de las muestras a lo largo del rayo), los resultados obtenidos están lejos de permitir la utilización de esta técnica para inspecciones interactivas del modelo.

	Esófago		Aorta	
<i>Viewport</i>	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles
<i>Muestras por voxel</i>	1	2	1	2
<i>Tiempo por imagen</i>	10 – 15 segundos	20 – 25 segundos	50 – 51 segundos	1 minuto 25 – 30 segundos

Tabla 4.4: Resultados obtenidos con ray-casting.

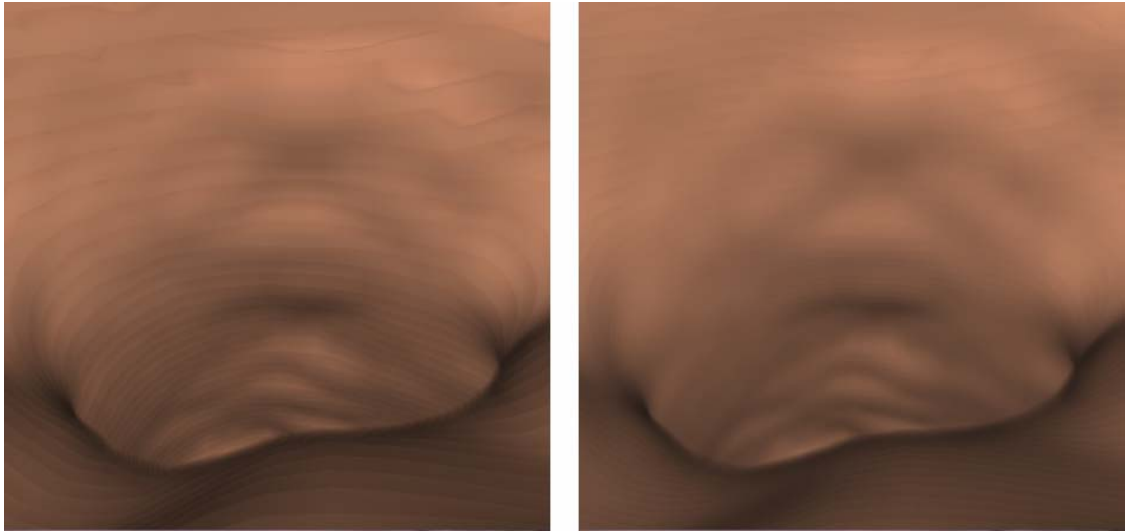


Figura 4.12: Imágenes del esófago obtenidas con ray-casting tomando 1 muestra (izq.) y 2 muestras por voxel (der.).

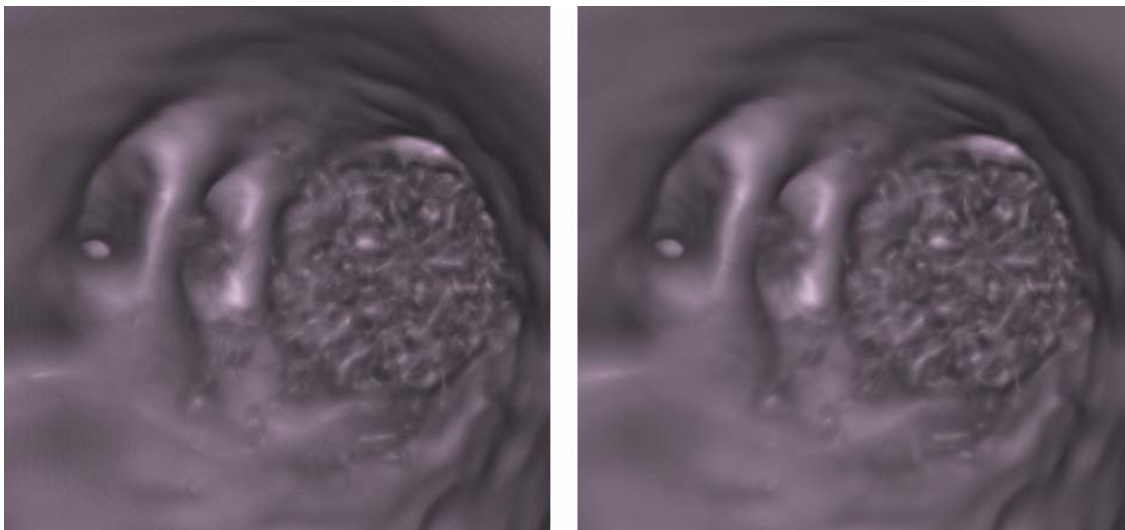


Figura 4.13: Imágenes con ray-casting de la aorta tomando 1 muestra (izq.) y 2 muestras por voxel (der.).

A diferencia de la visualización con isosuperficies, la técnica de *ray-casting* permite observar todas las estructuras anatómicas del volumen y variar su nivel de opacidad (ver figura 4.14). Este hecho permite observar al especialista la disposición de los diferentes elementos de interés respecto a las otras estructuras.

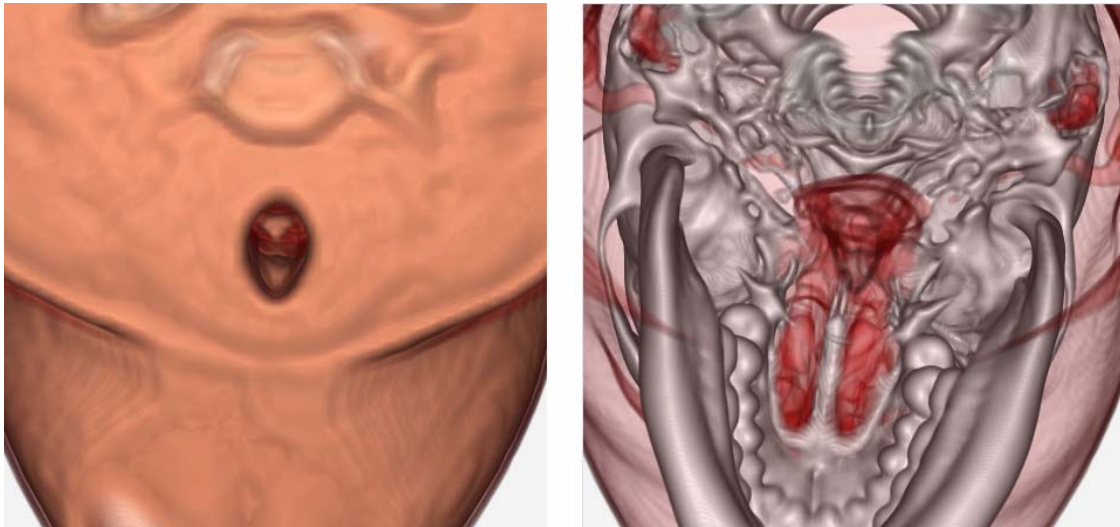


Figura 4.14: Imágenes obtenidas con ray-casting variando la opacidad de algunos tejidos.

En las tablas 4.5 y 4.6 se muestran los resultados obtenidos con texturas 3D resultantes de aplicar o no *shading*, es decir, pintando simplemente cada fragmento con el color resultante de la clasificación o aplicando el modelo de *Phong*. En las figuras 4.17 y 4.18 se puede observar que la calidad de las imágenes con *shading* es similar a las conseguidas con *ray-tracing*, y el tiempo de cálculo empleado para obtenerlas es mucho menor (del orden de décimas de segundo). En ambas tablas vemos que al doblar el número de muestras por *voxel*, el *frame rate* disminuye a la mitad.

	Esófago		Aorta	
<i>Viewport</i>	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles
<i>Muestras por voxel</i>	1	2	1	2
<i>Frame rate</i>	22 - 45 fps	11 - 23 fps	22 - 37 fps	13 - 19 fps

Tabla 4.5: Resultados obtenidos con texturas 3D sin *shading*.

	Esófago		Aorta	
<i>Viewport</i>	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles	464 x 448 píxeles
<i>Muestras por voxel</i>	1	2	1	2
<i>Frame rate</i>	10 - 20 fps	5 - 10 fps	10 - 16 fps	5 - 8 fps
<i>Frame rate (*)</i>	24 - 48 fps	12 - 25 fps	26 - 40 fps	13 - 20 fps

(*) Con el sistema de aceleración implementado. *Viewport* reducido de 256 x 256 píxeles.

Tabla 4.6: Resultados obtenidos con texturas 3D y *shading*.

La relación calidad – interactividad en el caso de las texturas 3D se comporta de la siguiente manera: sin aplicar *shading* se consiguen *frame rates* interactivos y menor calidad de imagen (ver figuras 4.15 y 4.16), mientras que con *shading* la calidad es mucho mayor pero el *frame rate* no alcanza niveles interactivos. Utilizando la técnica de aceleración descrita en el apartado 4.4, se obtienen *frame rates* interactivos con *shading*, consiguiendo mejorar la relación calidad – interactividad inicial.

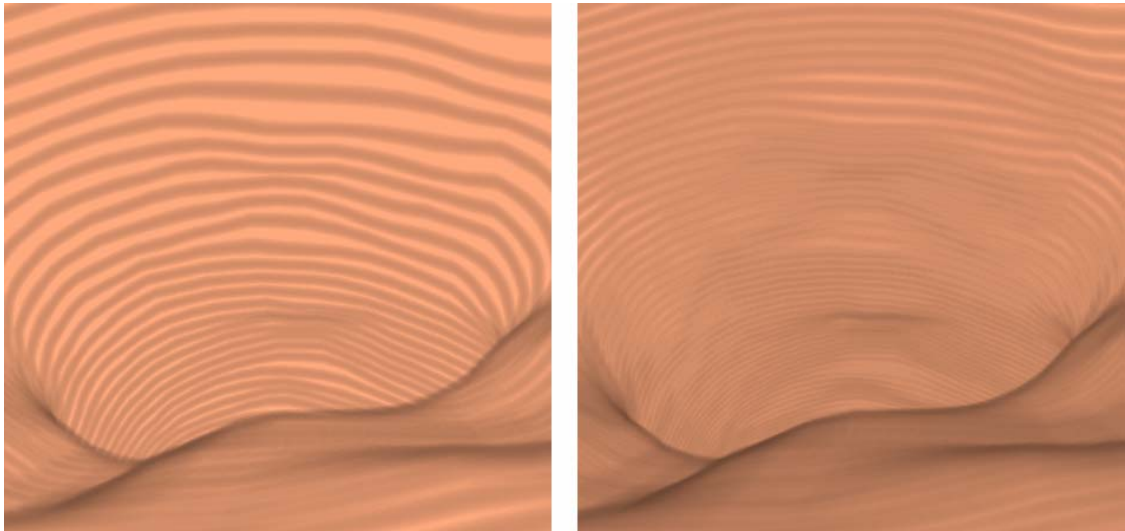


Figura 4.15: Imágenes obtenidas para el esófago con texturas 3D y sin shading. Resultado de tomar 1 (izq.) o 2 muestras por voxel (der.).

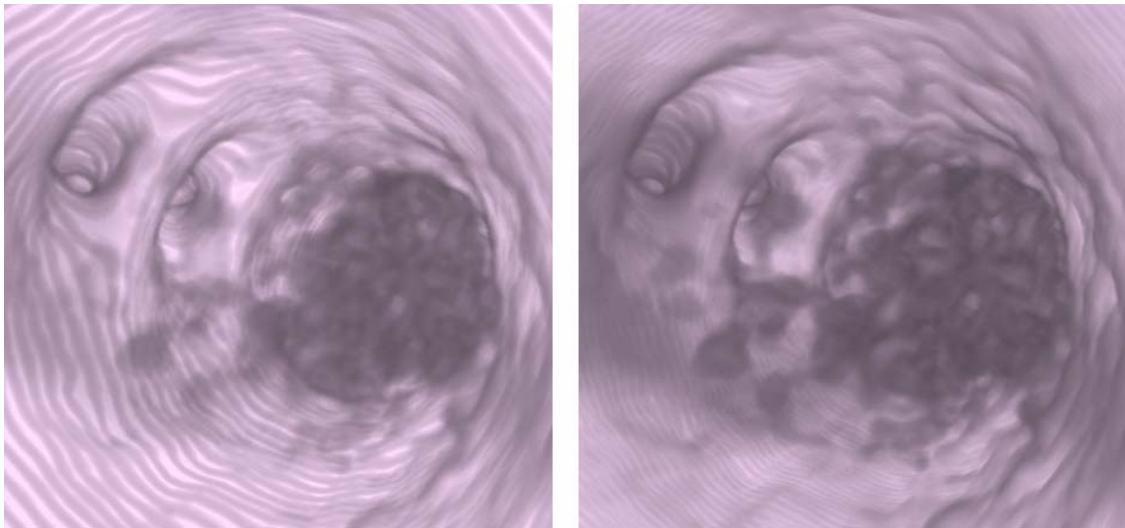


Figura 4.16: Imágenes obtenidas para la aorta con texturas 3D y sin shading. Resultado de tomar 1 (izq.) o 2 muestras por voxel (der.).

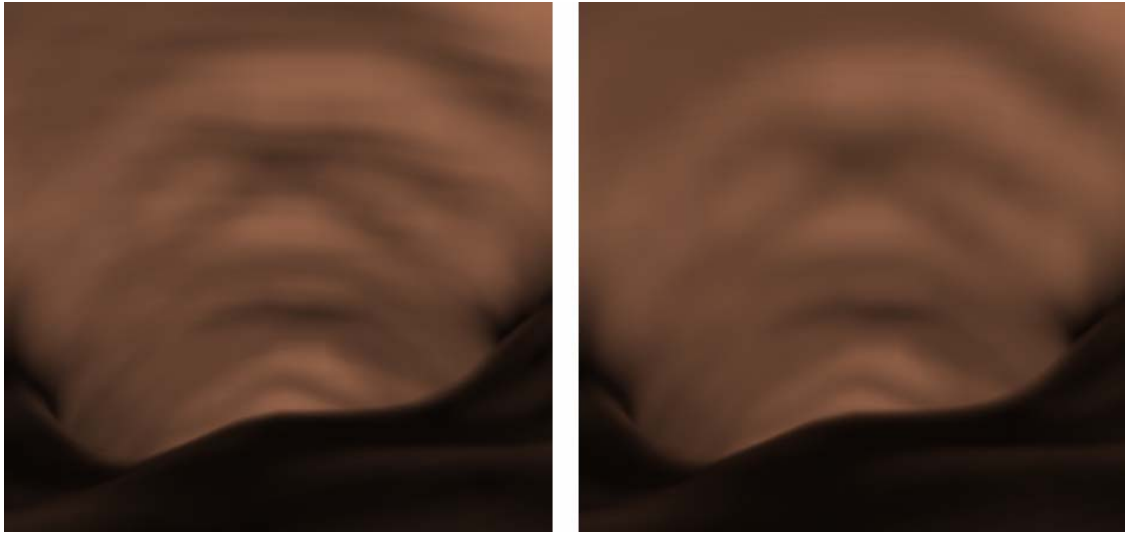


Figura 4.17: Imágenes del esófago con texturas 3D y shading. Resultado de tomar 1 (izq.) o 2 muestras por voxel (der.).

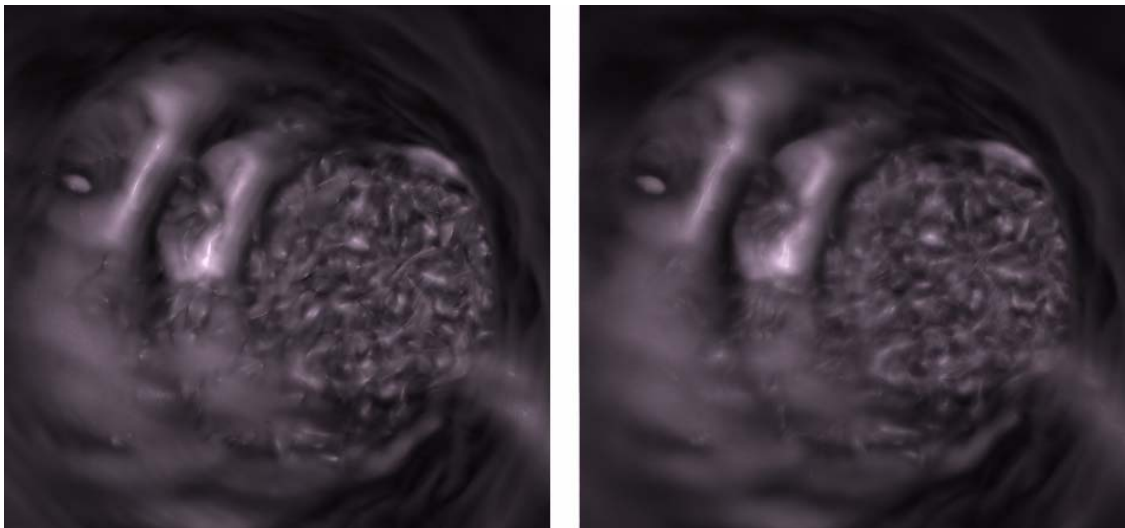


Figura 4.18: Imágenes obtenidas para la aorta con texturas 3D y shading. Resultado de tomar 1 (izq.) o 2 muestras por voxel (der.).

De la misma manera que utilizando la técnica de *ray-tracing*, con texturas 3D también se puede variar la opacidad de las diferentes estructuras anatómicas del volumen (ver figura 4.19). Tanto con una técnica como con otra, este efecto se consigue modificando la opacidad de los rangos de valores definidos en la función de transferencia utilizada en la etapa de clasificación (ver apartado 2.5).

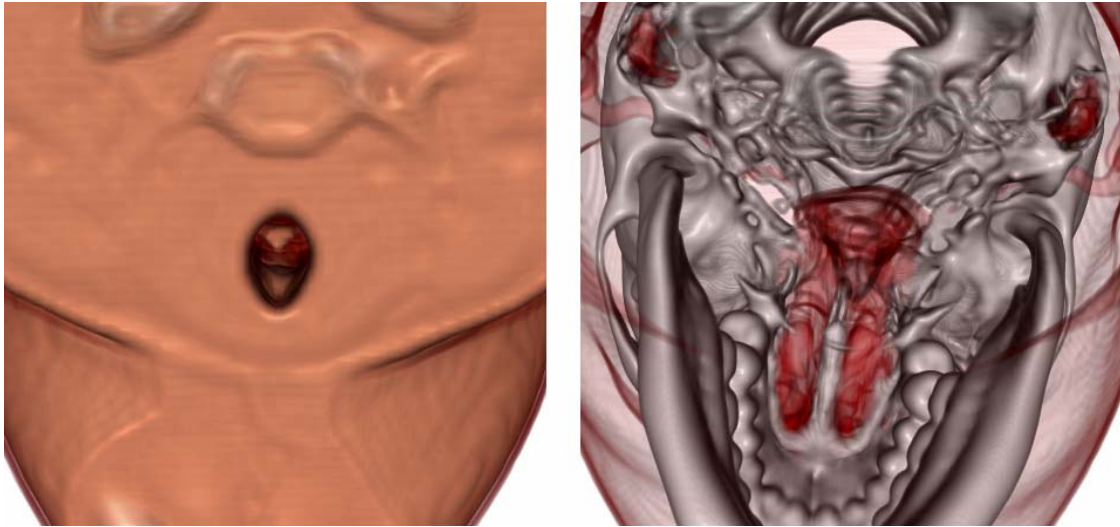


Figura 4.19: Imágenes obtenidas con texturas 3D variando la opacidad de algunos tejidos.

Una vez se han comparado los resultados obtenidos con las técnicas de visualización presentadas en el apartado 2.6 de esta memoria, se ha realizado un segundo conjunto de tests para evaluar el comportamiento de la relación calidad – interactividad conseguida con las técnicas presentadas en el apartado 4.4.

	Esófago	Aorta
<i>Viewport</i>	464 x 448 píxeles	464 x 448 píxeles
<i>Frame rate Modalidad 1</i>	5 – 8 fps	6 – 7 fps
<i>Frame rate Modalidad 2</i>	5 – 7 fps	6 – 7 fps
<i>Frame rate Modalidad 3</i>	5 – 10 fps	5 – 8 fps
<i>Frame rate Modalidad 4</i>	7 – 10 fps	7 – 9 fps

Tabla 4.7: Resultados obtenidos por las modalidades de adaptive sampling con shading y sin aceleración.

	Esófago	Aorta
<i>Viewport aplicación</i>	464 x 448 píxeles	464 x 448 píxeles
<i>Viewport reducido</i>	256 x 256 píxeles	256 x 256 píxeles
<i>Frame rate Modalidad 1</i>	14 – 20 fps	16 – 17 fps
<i>Frame rate Modalidad 2</i>	14 – 20 fps	16 – 17 fps
<i>Frame rate Modalidad 3</i>	12 – 25 fps	13 – 20 fps
<i>Frame rate Modalidad 4</i>	18 – 25 fps	17 – 23 fps

Tabla 4.8: Resultados obtenidos por las modalidades de adaptive sampling con shading y aceleración.

Las tablas 4.7 y 4.8 muestran los resultados conseguidos con las modalidades de *adaptive sampling* implementadas, sin aplicar y utilizando la técnica de aceleración. En ambos casos se aplican el *shading* al volumen ya que el objetivo que nos hemos marcado con las técnicas de *adaptive sampling* es el de mejorar la calidad de las imágenes conseguidas con texturas 3D; sin aplicar el *shading* la calidad de imagen es inferior.

Obsérvese que los tiempos conseguidos para las 4 modalidades implementadas son muy parecidos sin la técnica de aceleración, y que se consiguen *frame rates* interactivos al aplicarla con un *viewport* reducido de aproximadamente $\frac{1}{2}$ del original. Las modalidades 1 y 2 presentan siempre los mismos *frame rates* ya que visualizan el mismo número de planos, mientras que las modalidades 3 y 4 permiten obtener *frame rates* ligeramente superiores debido a que se recalcula el número de planos que se envían a pintar según la distancia entre Z_{near} y Z_{far} (ver apartado 4.4); para distancias pequeñas, el número de planos que se pintan en las modalidades 3 y 4 es inferior al utilizado en las modalidades 1 y 2.

Como se muestra en la figura 4.20, con las 4 modalidades implementadas se consigue mejorar la calidad obtenida sin aplicar *adaptive sampling*. Podemos observar que en la imagen obtenida con la modalidad 1 apenas se distinguen los *artifacts* debidos a la frecuencia de muestreo. Esto se debe a que en este caso es la modalidad que agrupa los planos a menor distancia del observador. Con la modalidad 2 se consigue una calidad parecida, si bien se pueden apreciar en mayor medida los *artifacts*. Las modalidades 3 y 4 presentan unos resultados similares, aunque de calidad inferior a la obtenida con las modalidades anteriores. Esto se debe a que en las modalidades 3 y 4 se pueden tener como máximo 2 muestras (planos) por *voxel*, mientras que en las otras modalidades el número de muestras puede ser mayor.

Como conclusión general observamos que los resultados obtenidos coinciden con los que se apuntaban en el capítulo 2 de esta memoria: en general, las técnicas de *Direct Volume Rendering* consiguen imágenes de mayor calidad que la visualización de isosuperficies, pero su nivel de interactividad es inferior. Con texturas es posible obtener imágenes de una calidad similar a las conseguidas con *ray-tracing* en menos tiempo.

Por otra parte, es posible reducir los *artifacts* en la visualización con texturas 3D y conseguir interactividad utilizando *adaptive sampling*. De las 4 variantes de *adaptive sampling* implementadas, las modalidades 1 y 2 son las que permiten obtener visualizaciones de mayor calidad, mientras que el tamaño óptimo del *viewport* reducido en la técnica de aceleración es $\frac{1}{2}$ del tamaño del *viewport* original (para tamaños superiores empeora la interactividad y para tamaños inferiores empeora la calidad).

Gracias al uso de las técnicas de optimización desarrolladas, nuestro sistema de endoscopias virtuales es capaz de obtener un nivel de interactividad semejante con texturas 3D al obtenido con las mallas de triángulos. Por tanto, se ha conseguido el propósito de obtener imágenes de calidad con *frame rates* interactivos.

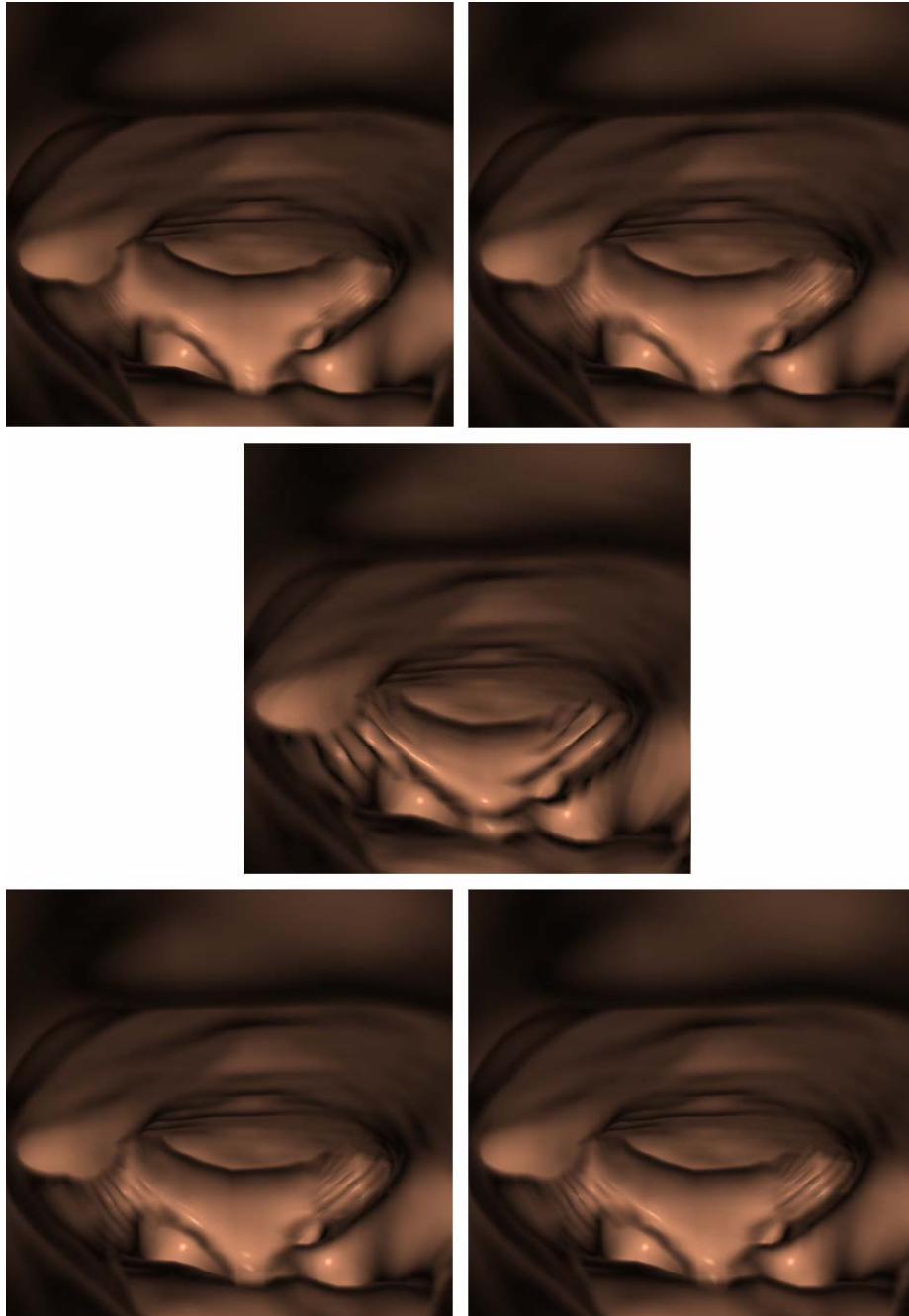


Figura 4.20: Comparación de la calidad obtenida sin aplicar (centro) y utilizando las modalidades de adaptative sampling implementadas. Modalidad 1 (arriba, izquierda), modalidad 2 (arriba, derecha), modalidad 3 (abajo, izquierda) y modalidad 4 (abajo derecha).

Capítulo 5

Conclusiones y trabajo futuro

El objetivo principal de esta tesis era la evaluación de los sistemas existentes de endoscopias virtuales, sus limitaciones, posibles soluciones y analizar el uso de las texturas 3D para poder realizar navegaciones interactivas de calidad. Para ello se han presentado los aspectos relacionados con la visualización e inspección de estructuras anatómicas tubulares.

Se ha llevado a cabo un análisis de las diferentes etapas que intervienen en la visualización de datos de volumen (capítulo 2). Se han descrito las características principales de los sistemas de endoscopia virtual existentes (capítulo 3), así como las principales limitaciones que presentan (apartados 2.6 y 4.2). Se han propuesto algunos métodos alternativos para superar esas limitaciones (apartado 4.4) y se han evaluado los resultados conseguidos (apartado 4.5).

La principal preocupación en el campo de las endoscopias virtuales es conseguir resultados de calidad sin que afecten al rendimiento del sistema. Para las técnicas utilizadas en la inspección de estructuras anatómicas, la calidad de las imágenes es inversamente proporcional al nivel de interactividad con el modelo, es decir, a mayor calidad, menor *frame rate*.

Se han analizado y evaluado diferentes propuestas para mejorar la relación calidad – interactividad. Se han implementado varias modalidades de la técnica de *adaptive sampling*, un método de aceleración con el *viewport* adaptativo y un sistema de detección de colisiones.

Respecto a las modalidades de *adaptive sampling* implementadas, todas ellas permiten aumentar la calidad de las imágenes finales, siendo las basadas en agrupar 2/3 de los planos a poca distancia del observador las que consiguen mejores resultados. Adicionalmente, el sistema implementado permite visualizar las estructuras del volumen con distintos niveles de transparencia, lo que resulta útil al especialista para observar la relación entre las estructuras anatómicas.

El sistema de detección de colisiones desarrollado cumple el propósito de no penalizar el rendimiento de la aplicación. Consiste en lanzar un rayo en la dirección de movimiento y encontrar la primera intersección con la superficie del conducto inspeccionado. Si no hay intersección o la distancia es superior a un margen de seguridad, se permite el movimiento.

Globalmente, se ha conseguido realizar endoscopias virtuales de gran calidad y de forma interactiva, con lo que se han alcanzado los objetivos planteados inicialmente.

El trabajo realizado se ha centrado en las maneras de mejorar la calidad e interactividad en la etapa de visualización de los datos de volumen. Como ámbitos de trabajo futuro, se puede implementar la detección de colisiones con la GPU o se pueden explorar nuevas técnicas que permitan aumentar la calidad de la visualización de isosuperficies, por ejemplo a base de incrementar la resolución de la malla al realizar *zooms*.

Visualizar estructuras poligonales ayuda en la realización de cálculos (volúmenes, distancias, etc.) en los que se requiere geometría. Por ello, otra línea de trabajo futuro puede ser la visualización conjunta de texturas e isosuperficies; en este sentido se piensa en técnicas que reconstruyan de forma local las isosuperficies (solo en las zonas visibles del volumen), realizando la composición de superficie y volumen con la GPU.

Acabar mencionando que sería interesante adaptar el sistema desarrollado al ámbito de la realidad virtual, ofreciendo los resultados con visualizaciones estereoscópicas e implementando los sistemas de navegación con el hardware específico para aplicaciones virtuales (sensores de movimiento, dispositivos hápticos, etc.).

Bibliografia

- [1] BARTZ, D.; Virtual Endoscopy in Research and Clinical Practice (STAR 4). In *Proc. of Eurographics State-of-the-Arts-Reports*. 2003.
- [2] BARTZ, D.; SKALEJ, M. VIVENDI - A Virtual Ventricle Endoscopy System for Virtual Medicine. In *Proc. of Symposium on Visualization*, pages 155–166,324, 1999.
- [3] ENGEL, K.; HADWIGER, M.; KNISS, J. M.; LEFOHN, A. E.; SALAMA, C. R.; WEISKOPF, D. Real-time volume graphics. In *ACM SIGGRAPH 2004 Course Notes*. ACM Press, New York, NY, 29. 2004.
- [4] GERING, D.; NABAVI, A.; KIKINIS, R.; HATA, N.; O'DONNELL, L.; GRIMSON, E.; JOLESZ, F.; BLACK, P.; WELLS, W. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. *J. Magn. Reson. Imaging*, 13:967–975, 2001
- [5] GOBBETTI, E., PILI, P., ZORCOLO, A., TUVERI M. Interactive Virtual Angioscopy. In *Proceedings IEEE Visualization*, pages 435-438. 1998.
- [6] HADWIGER, M., LANGER, C., SCHARSACH, H., BÜHLER, K. State of the Art Report 2004 on GPU-Based Segmentation. *Technical report, VRVis Research Center, Vienna, Austria*. 2004.
- [7] HONG, L., MURAKI, S., KAUFMAN, A., BARTZ, D., HE, T. Virtual Voyage: Interactive Navigation in the Human Colon. In *Proc. of SIGGRAP 1997 Conference*, pages 27-35, 1997.
- [8] HONG, L; KAUFMAN, A; WEI, Y; VISWAMBHARAN, A; WAX, M.; LIANG, Z. 3D Virtual Colonoscopy. *IEEE Symposium on Biomedical Visualization*, pages 26-32, 1995.
- [9] JOLESZ, F.; LORENSEN, W.; SHINMOTO, H.; ATSUMI, H.; NAKAJIMA, S.; KAVANAUGH, P.; SAIVIROONPORN, P.; SELTZER, S.; SILVERMAN, S.; PHILLIPS, M.; KIKINIS, R.. Interactive virtual endoscopy. *AJR*, 169, pages 1229–1235, 1997.
- [10] KRUGER, J.; WESTERMANN, R. Acceleration techniques for GPU-based volume rendering. *Visualization, 2003. VIS 2003. IEEE 19-24 Oct. 2003*, pages 287-292.
- [11] LACROUTE, P.; LEVOY M. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform. In *Proc. of SIGGRAPH 1994*, pages 451-458, 1994.

- [12] LENGYEL, J.; REICHERT, M.; DONALD, B. R.; GREENBERG, D. R. Real-time robot motion planning using rasterizing. *Computer Graphics* 24, pages 327-335. 1990
- [13] LEVOY, M. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(5), pages 29–37, May 1988.
- [14] LORENSEN, W. E.; CLINE, H. E. Marching cubes: a high resolution 3D surface construction algorithm. In *Proc. of SIGGRAPH 1987*, pages 163–169, 1987.
- [15] MAX, N. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2), pages 99–108, 1995.
- [16] STRAUSS, P.; CAREY, R. An Object-oriented 3D Graphics Toolkit. In *Proc. of ACM SIGGRAPH 1992*, pages 341-349, 1992.
- [17] VAN GELDER, A.; KIM, K. Direct volume rendering with shading via three-dimensional textures. In *Proc. of Symposium on Volume Visualization 1996*. San Francisco, California, U. S. 1996.
- [18] VILANOVA, A.; KÖNIG, A.; GRÖLLER, E. VirEn: Virtual Endoscopy System. *Machine Graphics & Vision*, 8(3), pages 469 - 487, 1999.
- [19] VINING, D. J.; STELTS, D. R.; AHN, D. K.; HEMLER, P. F.; GE, Y.; HUNT, G. W.; SIEGE, C.; McCORQUODALE, D. B.; SAROJAK, M. E.; FERRETTI, G. Freeflight: a virtual endoscopy system. Lecture Notes In *Computer Science*, vol. 1205. Springer-Verlag, London, pages 413-416. 1997
- [20] WESTOVER, L. Footprint evaluation for volume rendering. In *Proc. of SIGGRAPH 1990*, pages 367–376, 1990.
- [21] YOU, S.; HONG, L.; WAN, M.; JUNYAPRASERT, K.; KAUFMAN, A.; MURAKI, S.; ZHOU, Y.; WAX, M.; LIANG, Z. Interactive volume rendering for virtual colonoscopy. In Roni Yagel and Hans Hagen, editors. *IEEE Visualization 1997*, pages 433–346, 1997.
- [22] [Recurso en línea]. [Consulta: 16 julio 2007]. Disponible en <<http://www.postulate.org/segmentation.php>>

Anexo A

Para llevar a cabo los diferentes análisis comparativos realizados en la tesis, se ha desarrollado una aplicación capaz de realizar endoscopias virtuales que permite la exploración de estructuras anatómicas con las tres técnicas de visualización evaluadas.

La aplicación desarrollada abarca todas las etapas de un sistema de endoscopias virtuales, desde la generación del modelo de volumen a partir de los datos captados, pasando por la segmentación y clasificación de los datos, la visualización de las estructuras anatómicas, hasta el sistema de navegación para poder realizar la inspección. En este anexo se describen las características principales del sistema desarrollado y se presenta el código fuente de los *shaders* implementados.

A.1 Características del sistema de endoscopia virtual

Los datos de volumen se obtienen a partir de imágenes CT guardadas en formato *DICOM*. A partir de aquí el proceso a seguir es diferente según si la visualización se hace con mallas de triángulos o bien con las técnicas de *Direct Volume Rendering*.

Para la visualización con mallas de triángulos se realiza la segmentación del volumen seguida de la extracción de isosuperficies con el algoritmo de *Marching Cubes*. Para ello se dispone de dos métodos de segmentación semiautomáticos basados en las técnicas de *thresholding* y de *region-growing*, donde el usuario debe introducir un rango de valores o un conjunto de semillas respectivamente.

Para visualizar el modelo de volumen con *DVR*, la etapa de clasificación se realiza mediante funciones de transferencia, encargadas de asignar las propiedades ópticas a los datos. Las funciones que se han de aplicar a los modelos las define el usuario según los resultados visuales que pretenda conseguir. El sistema permite la modificación de sus parámetros de forma interactiva mostrando los resultados de los cambios en la visualización del volumen.

El interfaz de la aplicación ofrece al usuario cuatro vistas, tres de ellas corresponden a secciones 2D del modelo (axial, sagital y coronal) mientras que la cuarta presenta el modelo en una vista 3D (ver figura A.1). Se permite la obtención de imágenes tanto en proyección perspectiva como en proyección axonométrica, y es en la vista 3D donde se muestran los resultados de la visualización con texturas o *ray-casting*. Las imágenes obtenidas con la técnica de *visualización de isosuperficies* se muestran en una ventana aparte.

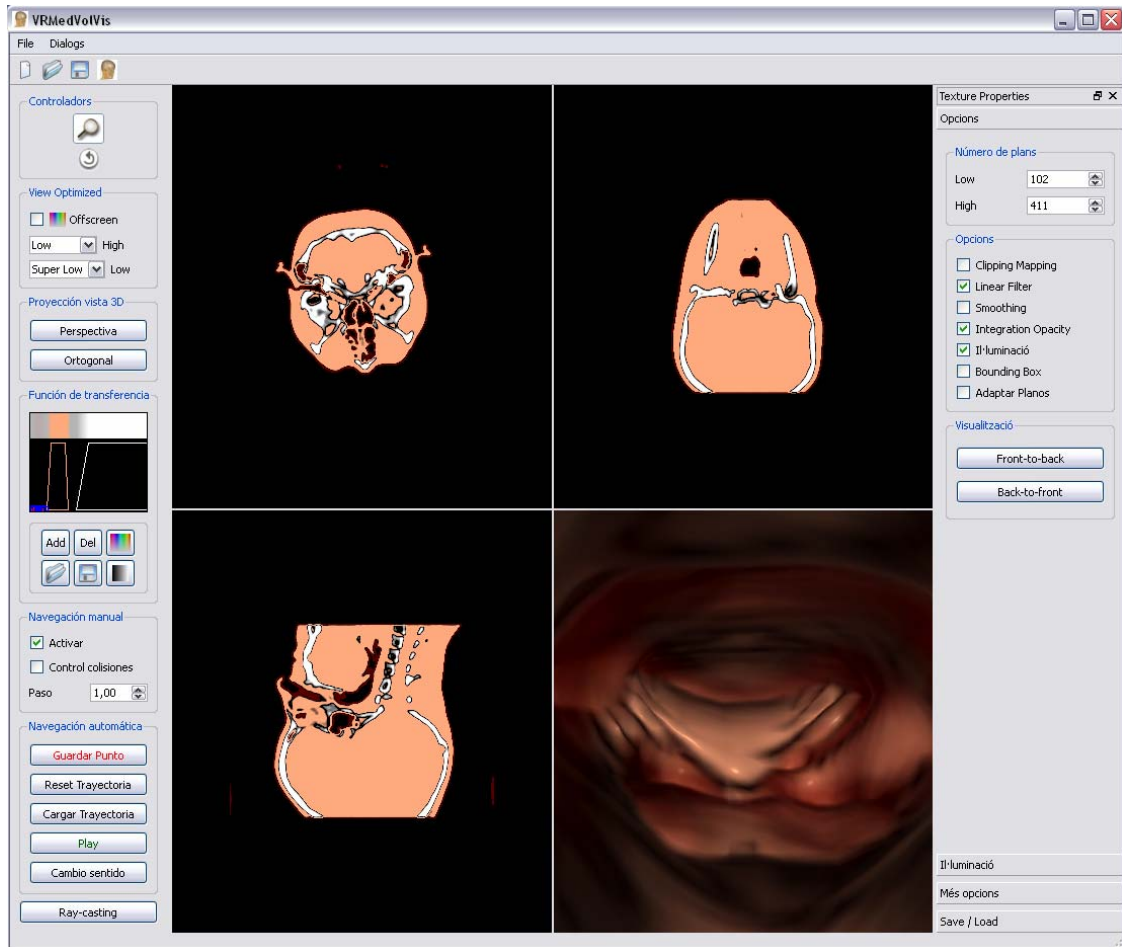


Figura A.1: Interfaz de la aplicación.

La aplicación permite al usuario visualizar las estructuras de datos como una malla de triángulos a la que se le puede aplicar *Gouraud shading* o *Phong shading*. El coloreado de *Gouraud* está implementado por software mientras que el de *Phong* se realiza en un shader para ejecutarlo con el hardware gráfico y reducir el tiempo de cálculo (ver apartado A.2).

El sistema de *ray-casting* está implementado por software y cuenta con la técnica de aceleración *early ray termination*, de forma que deja de recorrer el rayo cuando se alcanza una opacidad de 0.95. Este sistema de visualización se utiliza para obtener imágenes de mayor calidad en vistas concretas, ya que el tiempo que se invierte en conseguir cada imagen es elevado y no apto para la inspección interactiva del modelo.

La visualización con texturas 3D está implementada siguiendo la técnica *Voltx* [17] presentada por Allen Van Gelder y Kwansik Kim en 1996. Este método utiliza el concepto de texturación 3D con planos perpendiculares a la dirección de visión y permite realizar el *shading* del modelo obteniendo *frame rates* interactivos. Los cálculos de la iluminación se realizan en tiempo de ejecución en un *shader* (ver apartado A.2), para aprovechar la potencia de cálculo de la GPU.

La aplicación permite seleccionar al usuario el número de planos con los que se visualiza el modelo de volumen, el orden en el que se envían a pintar y el modo en que se realiza la composición de color: *back-to-front* o *front-to-back*. Además, para mejorar la calidad de las imágenes, se han implementado 4 variantes de *adaptive sampling* que permite distribuir el número de planos a pintar entre el Z_{near} y el Z_{far} , eliminando los *artifacts* originados por la frecuencia de muestreo.

El sistema de endoscopias virtuales incorpora los tres modelos de navegación analizados en el apartado 4.4 de la memoria: navegación manual, navegación automática y navegación guiada. En la navegación automática, la aplicación ofrece al usuario la posibilidad de visualizar la animación en ambos sentidos del recorrido definido por los *key frames* (desde el punto inicial al punto final y viceversa). De esta forma, se pueden inspeccionar los órganos desde dos puntos de vista diferentes. Además, el sistema ofrece la posibilidad de cargar y recorrer las trayectorias definidas por el usuario.

A.2 Shaders utilizados en el sistema

En este apartado se incluye el código fuente de los *shaders* utilizados en la aplicación.

• Iluminación de Phong para isosuperficies

```
varying vec3 Normal;
varying vec3 EyeDir;
varying vec3 LightVec;

void main(void) {
    gl_Position = ftransform();
    Normal      = gl_NormalMatrix * gl_Normal;
    vec4 pos    = gl_ModelViewMatrix * gl_Vertex;
    EyeDir      = - pos.xyz;
    vec4 light  = gl_ModelViewMatrix * gl_LightSource[0].position;
    LightVec    = light.xyz;
}
```

Figura A.2: Vertex shader para calcular el Phong shading.

```

varying vec3 Normal;
varying vec3 EyeDir;
varying vec3 LightVec;

void main()
{
    vec3 N = Normal;
    if (!gl_FrontFacing) N = -N;

    vec4 color = vec4(0,0,0,1);

    N = normalize(N);
    vec3 V = normalize(EyeDir);
    vec3 L = normalize(LightVec);
    vec3 R = reflect(-L,N);

    float NdotL = max(dot(L,N), 0.);
    float VdotR = max(dot(V,R), 0.);

    color += NdotL * gl_FrontMaterial.diffuse *
            gl_LightSource[0].diffuse;

    color += gl_FrontMaterial.specular *
            gl_LightSource[0].specular *
            pow(VdotR,128.0);

    color += gl_FrontMaterial.ambient *
            gl_LightSource[0].ambient;

    gl_FragColor = color;
}

```

Figura A.3: Fragment shader para calcular el Phong shading.

- *Composición del color para texturas 3D*

```

void main()
{
    gl_Position = ftransform();
    gl_TexCoord[0] = gl_TextureMatrix[0] * gl_MultiTexCoord0;
    gl_ClipVertex = gl_ModelViewMatrix * gl_Vertex;
}

```

Figura A.4: Vertex shader para calcular el Phong shading.

```

uniform sampler3D VolTex;
uniform sampler2D PalTex;
uniform vec2 PalRes;
uniform vec3 LightDirection;
uniform vec3 DimVoxel;
uniform vec3 IncrTex;
uniform float GradModulation;
uniform float SamplingDistance;
uniform float GradContribution;
uniform float FactorPaleta;
uniform float BlendMode;
uniform float VisMode;
uniform float FactorEscala;
uniform float AlphaTest;

void main(){

    float v = texture3D( VolTex, gl_TexCoord[0].xyz ).a;
    float valor = v * (PalRes.x * PalRes.y - 1.0)*FactorPaleta;
    vec2 coord;

    coord.y = floor( valor / PalRes.x );
    coord.x = valor - ( coord.y * PalRes.x );
    coord.x /= (PalRes.x - 1.0 );
    coord.y /= ( PalRes.y - 1.0 );

    vec4 material = texture2D (PalTex, coord);
    vec3 color = material.rgb;

    float opacidad = min (1.0, material.a*SamplingDistance);

    if (opacidad > 0.0){
        vec3 coord = gl_TexCoord[0].xyz + vec3(IncrTex.x, 0., 0.);
        vec3 vM, vM;
        vM.x = texture3D (VolTex, coord).a;
        coord = gl_TexCoord[0].xyz - vec3(IncrTex.x, 0., 0.);
        vM.x = texture3D (VolTex, coord).a;
        coord = gl_TexCoord[0].xyz + vec3(0., IncrTex.y, 0.);
        vM.y = texture3D (VolTex, coord).a;
        coord = gl_TexCoord[0].xyz - vec3(0., IncrTex.y, 0.);
        vM.y = texture3D (VolTex, coord).a;
        coord = gl_TexCoord[0].xyz + vec3(0., 0., IncrTex.z);
        vM.z = texture3D (VolTex, coord).a;
        coord = gl_TexCoord[0].xyz - vec3(0., 0., IncrTex.z);
        vM.z = texture3D (VolTex, coord).a;

        vec3 N = vM - vM;
        N = N /DimVoxel;
        float mag = length (N);
        N = N/mag;
        float NL = abs(dot (LightDirection, N));
    }
}

```

```

float pond = GradModulation * mag;
opacidad = opacidad * (1.0 - GradModulation + pond);

if (mag > GradContribution){
    vec3 Od = color;
    color = gl_FrontLightModelProduct.sceneColor.rgb * Od;
    color+= gl_FrontLightProduct[0].diffuse.rgb * Od * NL;
    color+= gl_FrontLightProduct[0].specular.rgb * Od *
        pow(NL, gl_FrontMaterial.shininess);
}
}

if (VisMode == 3.0){

float valEsc = clamp (v*FactorEscala, 0.0, 1.0);

    if (AlphaTest == 1.0){
        gl_FragColor = vec4(valEsc, valEsc, valEsc, v);
    }
    else
    {
        if (BlendMode == 2.0){
            gl_FragColor = vec4(valEsc, valEsc, valEsc,
                opacidad);
        }else{
            gl_FragColor = vec4(color,opacidad);
        }
    }
}
else
{
    gl_FragColor = vec4(color,opacidad);
}
}
}

```

Figura A.5: Fragment shader para calcular el Phong shading.