Computers & Graphics 133 (2025) 104453

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

# Hybrid visualisation of medical image: Surface and volume rendering☆

Belén Palma [a,*], Pablo Casanova-Salas [b], Jesús Gimeno [b], Sergio Casas-Yrurzum [b]

[a] Institute of Robotics and Information and Communication Technologies (IRTIC), Universitat de València, Paterna, 46980, Spain
[b] Computer Science Department, University of Valencia, Valencia, 46010, Spain

## ARTICLE INFO

*Keywords:*
Computer graphics
Real-time
Virtual reality
Healthcare visualisation

## ABSTRACT

Medical imaging plays a crucial role in diagnosis and surgical planning, but accurately merging segmented 3D models with volumetric medical data remains a significant challenge. Traditional surface-based 3D models provide a clear structural representation but may introduce segmentation inaccuracies, whereas Direct Volume Rendering (DVR) maintains all image details but lacks explicit surface definition. We present a novel hybrid rendering algorithm that seamlessly integrates Per-Pixel Linked Lists for accurate transparency sorting of 3D meshes with DVR techniques to ensure a smooth and precise blending of both data types. This approach captures the advantages of both methods—preserving the structural clarity of surface models while maintaining the depth and density information from the medical image. Our algorithm operates in real time and is compatible with virtual reality, enhancing spatial perception through stereoscopic visualisation. By allowing direct interaction with both segmented models and volumetric data, this technique reduces reliance on radiologists' reports and improves surgical planning. Performance evaluations confirm real-time rendering, and an expert-based usability study further supports the method's relevance for advanced medical visualisation applications.

## 1. Introduction

Medical imaging plays a fundamental role in disease diagnosis, treatment planning, and surgical decision-making. However, traditional visualisation methods rely on the analysis of 2D slices, requiring specialists to mentally reconstruct 3D structures. This process is complex, cognitively demanding, and typically restricted to radiologists, limiting direct interaction with full volumetric data for other medical professionals, such as surgeons. The study designed by Wake et al. [1] highlights the fact that even experienced surgeons often struggle to translate 2D imaging data into accurate 3D interpretations. Furthermore, the use of segmented 3D models in the surgical planning process tends to reduce the time required by clinical surgeons to complete tasks that require spatial reasoning skills [2]. In this context, Necker et al. [3] demonstrated that interactive 3D visualisation significantly improves surgeons' confidence in anatomical interpretation when compared to conventional 2D precomputed volume rendering. Whereas segmented 3D models offer a more intuitive representation of anatomical structures, they often lack the full detail present in the original medical images. Automated segmentation can introduce inaccuracies or overlook subtle but clinically significant features [4].

To address these limitations, we propose a novel hybrid rendering method that integrates the strengths of Per-Pixel Linked Lists for accurate transparency sorting of 3D meshes and Direct Volume Rendering (DVR) for detailed visualisation of volumetric medical images. Our approach seamlessly blends these two techniques, preserving both the structural clarity of segmented 3D models and the rich density information from medical imaging studies. Unlike existing methods that rely solely on surface rendering or volumetric visualisation, our algorithm ensures a precise fusion of both data sources, maintaining spatial coherence and enabling an intuitive exploration of medical cases. The key advantage of our system is its ability to provide a highly detailed, yet easily navigable, visualisation for non-expert users. The segmented 3D models serve as a guide for intuitive interaction, allowing users to inspect anatomical structures with familiar spatial references. At the same time, our method dynamically integrates the original medical image data into the visualisation, enhancing depth perception and ensuring that critical details, such as variations in tissue density, tumour boundaries, and anatomical anomalies, are preserved and easily interpretable. By leveraging virtual reality (VR), the system further enhances spatial understanding through stereoscopic vision, allowing physicians to navigate complex anatomical structures with greater precision and confidence.

This hybrid visualisation technique reduces reliance on radiologists' reports by providing a direct, interactive method for exploring medical

☆ This article was recommended for publication by Stefan Bruckner.

* Corresponding author.
*E-mail address:* belen.palma@uv.es (B. Palma).

https://doi.org/10.1016/j.cag.2025.104453
Received 21 April 2025; Received in revised form 1 October 2025; Accepted 8 October 2025
Available online 11 October 2025

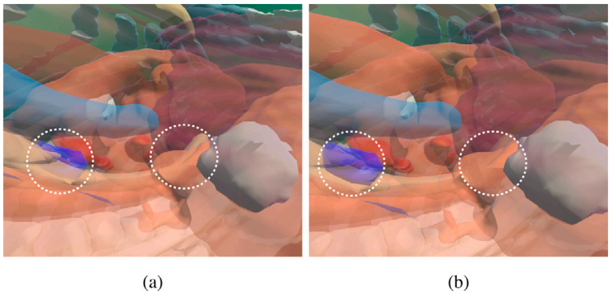


**Fig. 1.** Example of a depth-sorting problem in scenes with multiple transparent 3D objects. Fig. 1(a) shows the correct rendering depth-sorting: left circle highlights the blue tumour is rendered behind the flesh-coloured tissue, whereas the right circle points out the tissue in the middle of the orange and maroon tissues. In Fig. 1(b), there is a visual aberration in the depth-sorting: the blue tumour is rendered above the flesh-coloured tissue, whereas the right highlighted area shows that the principle tissue is rendered behind the marron one.

images in a way that is both accurate and accessible. Unlike traditional 2D viewers or purely segmented 3D models, our approach ensures that no clinically relevant information is lost while maintaining the usability needed for real-time medical decision-making.

This paper is structured as follows: Section 2 reviews related work in medical visualisation and rendering techniques. Section 3 presents the details of our novel hybrid rendering algorithm and its implementation. Section 4 evaluates the system's effectiveness through clinical case studies, focusing on visualisation accuracy and real-time performance. Finally, Section 5 discusses conclusions and future research directions.

## 2. Related work

In the work of Abhari et al. [2], the authors evaluate different visualisation methods for depth perception in medical image analysis and 3D meshing. Most users prefer semi-transparent visualisation methods for 3D model surfaces. In addition, these techniques improve user accuracy in completing tasks and increase user confidence. However, real-time rendering of multiple semi-transparent surfaces can cause sorting problems in common computer graphics algorithms, especially in cases where mesh intersection or self-sorting artefacts occur. When assessing tumours, it is possible to find complex 3D structures where tissues intersect or are surrounded by other viscera. In such cases, determining the correct rendering order of transparent 3D meshes becomes unfeasible, leading to visual artefacts that change dynamically with slight variations in the position and orientation of the camera. Fig. 1 shows the visual effect of a sorting problem during inspection of a 3D model, with semi-transparency, generated from a medical imaging study. Fig. 1(a) shows the correct initial rendering, but after a slight advance of the camera towards the centre of the 3D model, the sorting problem can be seen in Fig. 1(b). This abrupt change in the image can cause visual discomfort to the user and affect the spatial comprehension of the 3D models in the scene. For this reason, we consider it important to use *Order Independent Transparency* (OIT) techniques to define an exhaustive drawing order according to the depth of the objects.

### 2.1. Order-independent transparency

The Order-Independent Transparency techniques provide a structured solution for rendering transparent surfaces without requiring a predefined draw order. These methods help prevent visual artefacts, which are common in complex 3D scenes involving intersecting or overlapping geometry. The existing literature outlines a variety of algorithms that address transparency blending at the pixel level. Each approach is designed to balance different trade-offs, such as rendering performance, visual accuracy or memory usage, depending on the specific requirements of the rendering scenario.

The *Depth Peeling* algorithm [5] enables the rendering order of elements to be determined based on a predefined set of layers within the scene. This allows depth information to be retrieved for each layer individually, rather than only for the object nearest to the camera. However, this approach can significantly increase rendering time, as it requires a separate rendering pass for each depth layer. Moreover, its effectiveness is constrained by the number of layers, either predetermined by the algorithm or manually defined by the user. This poses challenges in complex scenes where multiple elements are densely packed within small spatial regions, as previously illustrated. Resulting visual ordering errors can confuse users without expertise in computer graphics, such as clinicians, who may misinterpret the output without understanding the rendering artefacts involved. As an improvement to this algorithm, the *Dual Depth Peeling* technique [6] is proposed, which can reduce the number of passes by N/2 for N layers. This improved version allows simultaneous peeling in both directions: from the camera to the background and from the background to the camera. Despite this optimisation, both methods may still store elements at different depths within the same layer if no intermediate geometry occludes them, which reduces depth-sorting precision.

Another prominent algorithm within the OIT techniques is weight-based sorting, known as *Weighted Blended Order-Independent Transparency* [7]. This approach reduces the computational cost of rendering as it can perform the sorting in a single pass without the need for depth-based sorting. This algorithm proposes the use of weights in the translucency equation that runs through the scene. In this case, the aim is to estimate the occlusion for each depth as a function of the transparency of the rest of the objects considered. This algorithm is not computationally or memory-intensive, but its results vary greatly depending on the type of scene being rendered. In cases where there are close clusters of several transparent elements of different colours, the rendering quality decreases. In addition, the assignment of weights in the blending algorithm needs to be adjusted by an expert for each scene according to the distribution of elements if quality sorting is to be achieved. Given the complexity of the mesh distribution of medical segmented 3D models and the need for accuracy in any depth range, this method cannot be used for our purposes.

For medical image visualisation and inspection, maintaining the accuracy of the final colour blend is essential to preserve the integrity of the underlying medical data. However, this requirement often entails a high computational cost, which can hinder the ability to sustain real-time frame rates, particularly in immersive visualisation environments, where low refresh rates may compromise usability. To address this, we adopted *Order Independent Transparency with Per-Pixel Linked Lists* [8] as the baseline for OIT. This method operates at the fragment level, analysing and sorting all object depths linked to each pixel. While it requires two rendering passes per frame, regardless of scene complexity, it remains compatible with real-time performance. Its precise, per-fragment depth sorting avoids layer-based grouping, making it particularly suitable for accurate blending with volumetric medical image data. The full blending algorithm is detailed in Section 3.

### 2.2. Direct volume rendering

Volume rendering techniques enable the visualisation of 2D medical image slices as a continuous 3D volume by simulating light absorption along rays cast through the voxel data [9].

Direct Volume Rendering (DVR) enables a direct mapping of volumetric medical data by casting rays through the voxel grid, accumulating colour and opacity values based on optical models. Unlike methods that rely on intermediate geometric primitives, DVR preserves the full internal structure of the volume, producing high-quality results. The method simulates light absorption along rays aligned with the camera's perspective, allowing detailed inspection of the data. The ease of ray tracing implementation on modern graphics cards has greatly improved the applicability of this method. For these reasons, this technique has become a reference for medical image rendering [10].

To improve the quality of the volume data visualisation and speed up the analysis phase, it is possible to find transfer functions applied on the volume data [10]. Moreover, multidimensional transfer functions help to better reveal the internal information of the volumetric data. In the literature, it is possible to find implementations of manual transfer functions where the user must first know the characteristics of the data. Shape analysis [11] or data volume gradient [12] are some examples based on physical characteristics. References can also be found in the literature that fully or partially automate the transfer function using statistical methods such as Kernel Density Estimation (KDE) [13] or Gaussian Mixture Model [14].

These methods enhance volumetric inspection by improving visual differentiation of internal structures. However, they remain focused on voxel data alone and do not explicitly encode anatomical boundaries or support high-level interaction. Although advanced voxel-based approaches, such as Split-Voxel technique [15], can preserve discontinuities and support transparency, they are often limited when precise spatial relationships, quantitative analysis, or user-guided interaction are required. For such tasks, mesh-based representations offer complementary benefits. This motivates the integration of both data types through hybrid rendering techniques.

### *2.3. Blending 3D models & volumetric medical image*

There are several solutions available in the literature that allow the rendering of medical images and 3D models with the use of OIT techniques. Visualisation Toolkit (VTK) [16], the well-known graphics library specialised in rendering scientific data, allows medical images to be rendered using *Direct Volume Rendering* (DVR) and the OIT *Depth Peeling* algorithm. A notable limitation of VTK's approach is its reliance on a fixed number of depth layers for transparency sorting, which does not automatically adapt to the specific depth requirements of each dataset. As a result, the user is often required to manually adjust the layer count and pre-analyse the model before rendering, leading to inefficiencies and potentially inaccurate results. In contrast, our method overcomes this issue by automatically covering the depth characteristics of each dataset, ensuring smoother and more accurate blending of volumetric and mesh data without manual intervention or extra computational cost.

It is also possible to find other methods based on layer generation but using other reference parameters such as cumulative opacity [17] or pre-classified tissue importance indices to guide the accumulation process [18]. However, these approaches can be inherently slow, as seen in Section 2.1. Merten et al. [19] present an alternative method based on *Indirect Volume Rendering* (IVR) and OIT *Linked List* algorithm to achieve joint rendering of information from PET and CT studies of the same case. This approach requires two rendering passes regardless of the complexity of the scene. Nevertheless, the Indirect Volume Rendering technique reflects only surfaces with a specific field value, resulting in a final view similar to rendering 3D mesh surfaces.

The application of per-pixel linked lists extends to volumetric detail mapping, as demonstrated by Patel et al. [20]. Their method for instant convolution shadows employs per-pixel linked lists to construct a Layered Depth Image (LDI) in a single geometry pass, which is then used to determine volumetric interior/exterior relationships during ray traversal. While focused on shadow computation rather than hybrid rendering, this approach highlights the effectiveness of per-pixel linked lists for capturing complex surface-depth information on the GPU.

Following the approaches discussed above, we propose the use of *Real-Time Concurrent Linked List* method [21], focused on Order-Independent Transparency for the 3D meshes, in conjunction with *Direct Volume Rendering* (DVR) [9] for obtaining a 3D volume of medical image. With the blending of these previous techniques, the number of rendering passes is independent of the complexity of the scene. In addition, it is possible to visualise the information from the medical image directly in 3D space. The stored pixel density of the clinical study is used as input colour in the blending algorithm. To the best of our knowledge, no existing algorithm combines *Direct Volume Rendering* with the OIT *Linked List* technique. The core of our proposed algorithm focuses on efficiently and accurately blending collections of continuous (medical image) and discontinuous (3D meshes) data in space.

Furthermore, we propose a virtual reality-based visualisation environment to showcase this new rendering method for medical images and 3D models. The use of stereoscopic rendering systems improves spatial understanding compared to MRI or 3D models themselves on a conventional 2D monitor [5]. In tumour response assessment and surgical planning, the study of the spatial relationships between the tissues involved is of great importance. This is an ideal case study for the analysis of our proposal. In addition, we consider this medical field to be of great interest as cancer is one of the main causes of premature mortality worldwide [22]. Virtual reality enables unrestricted inspection of 3D models, without the physical limitations of 3D printed models. Within virtual inspection environments, users can easily adjust the model's dimensions and visual appearance. The user can position various types of case-related data in hypertext format around the 3D model, allowing for integrated access to a large volume of data. Consequently, users can access a substantial amount of information within a unified environment. Finally, collaborative virtual environments can extend this framework by enabling shared surgical planning sessions. These platforms remove spatial barriers and support the inclusion of specialised simulation tools, such as trocar placement planning for robotic-assisted surgery (RAS) [23].

## 3. Implementation

Following careful consideration of all requirements reviewed in previous sections, and with the aim of creating an efficient algorithm for the virtual reality platform, we have designed an algorithm based on the Real-Time Concurrent Linked List [21] approach for managing Order-Independent Transparency and Direct Volume Rendering [9]. This method provides an accurate representation of the colour and transparency of the segmented 3D models and the 3D volume of the medical image in the scene. Our proposed system only requires the two mandatory passes of the linked-list-based OIT technique, since both the ordering of the OIT lists and the final colour blending using the DVR are performed jointly. For the design of this algorithm, shaders from the Unity game engine and the Direct3D11 graphics API, based on DirectX11, have been used. This 3D graphics API allows atomic operations to be performed, guaranteeing access to variables shared by multiple threads without restricting the order in which operations are performed in concurrent memory. It also includes predefined functions that are more efficient than using traditional mechanics, such as a global in-memory counter as opposed to the *IncrementCounter()* function [21].

Overall, the rendering algorithm performs 3 main steps for each final frame:

1. Store the 3D mesh information in GPU buffers.
2. Sort the 3D mesh information by depth.
3. Blend the information from the 3D meshes and the 3D texture of the medical image based on the DVR sampling technique.

The algorithm requires two global buffers in the GPU where all the information related to the 3D objects is stored: The "head pointer" buffer and the "node" buffer. The node buffer contains the nodes of the linked list following a data structure containing the following fields: colour, depth, position in the world and pointer to the next node in the list. The head buffer is an array of pointers of the same size as the frame buffer, pointing to the initial node of the generated linked list for

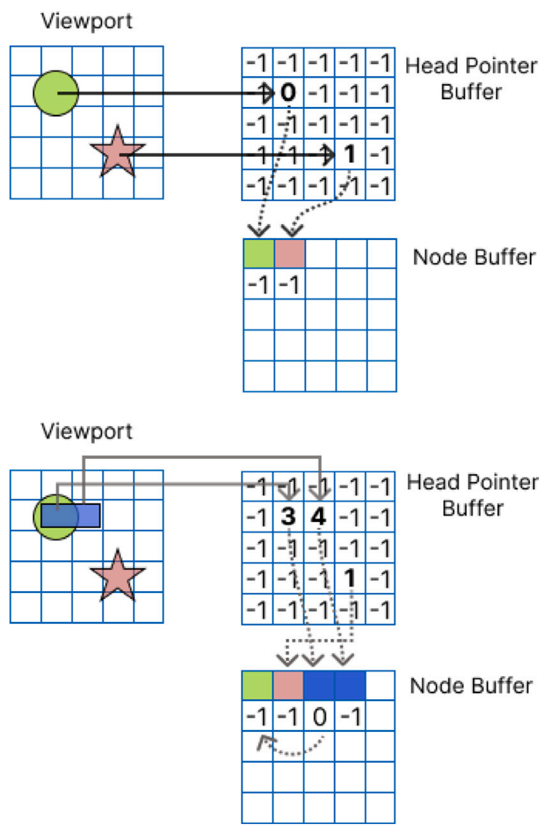



**Fig. 2.** Schematic diagram explaining the registration system of the objects to be rendered in the node buffer and the head pointer buffer. For clarity, only colour and pointer information are shown in the node buffer.

each screen pixel. An automatic incremental counter is used to manage storage within the node buffer, ensuring that the next available position can be identified efficiently. Fig. 2 schematically illustrates how data is inserted into the buffers during the rendering of overlapping 3D meshes. In order to facilitate understanding of the buffer filling process, the node buffer data structure has been simplified. Each entry stores the object's representative colour at the current viewport pixel and a pointer to the next node in the list. In the upper part of the figure, most cells in the head pointer buffer are initialised with negative values, indicating an initial state where no overlapping geometry is present. The lower section shows how new nodes are inserted and linked when overlaps occur. A key aspect is the update of head pointer entries corresponding to the blue rectangle. These now point to the closest mesh surfaces to the camera, which act as the first visible layer and occlude any geometry behind them. Examining the blue nodes in the node buffer reveals that they point to the previous entries that originally occupied the head pointer positions in the last rendering cycle, effectively maintaining the linked list structure.

### 3.1. Main rendering cycle

Each 3D mesh in the scene has an associated shader, which operates in the local spatial context of the mesh geometry. This shader is responsible for recording all the information in the head buffer and node buffer structures. All relevant spatial and visual attributes are stored to ensure that the data from different objects can be consistently aligned and blended during the global rendering pass. Instead of writing the mesh colour directly to the render target, the shader performs the following steps:

1. Query the address (P′) of the next free location in the node buffer.
2. Perform an exchange between the node address (P′) obtained in step 1 and the original pointer (P) in the head buffer for the position of the examined pixel. In this way, the pointer associated with the head pointer buffer points to the node we are currently working on, building a linked list.
3. Fill in the data structure information for each entry in the node buffer: colour, depth and position in the world. Use the value of the previous node (P) as a pointer to the next node.
4. Write the complete data structure to the node address (P′ ) obtained in the first step.

At the end of the rendering phase, all the structures of the scene will have been registered in the node buffer by using linked lists according to the pixel in which they are located. From this point on, it is possible to traverse the list associated with each pixel by accessing the initial pointers in the head pointer buffer. The use of atomic functions allows the safe treatment of the data in the parallel processing performed by the GPU. The process can be reflected in Algorithm 1.

**Algorithm 1:** Pseudocode of 3D mesh registration in Head Buffer and Node Buffer

**Data:** HeadPointerBuffer, NodeBuffer, Colour, ScreenParams
**Result:** HeadPointerBuffer, NodeBuffer

$pixelIndex \leftarrow vertex.y \times ScreenParams.x + vertex.x$;
$P' \leftarrow NodeBuffer.IncrementCounter()$;
$P \leftarrow 0$ ;
$InterlockedExchange(HeadPointerBuffer[pixelIndex], P', P)$;
$fragNode \leftarrow$
$NodeStructure(Colour, vertex.z, vertexWorldPos, P)$;
$NodeBuffer[P'] \leftarrow fragNode$;

Although atomic operations allow data to be processed in parallel threads, it is not possible to guarantee the order of execution of the threads in hardware. For this reason, it is necessary to apply a sorting algorithm based on the depth recorded for each node to ensure the correct order of the lists. The sorting algorithm is executed in a second, and final, shader when the camera has completed rendering. From this point on, all information related to the meshes is stored and sorted accurately in the node buffer. This information is used in conjunction with the medical image to be displayed by direct volumetric rendering.

The DVR algorithm operates globally from the camera's perspective, traversing the full volume defined by the medical image. It is based on a 3D texture derived from the collection of medical image slices and runs as a separate rendering pass from the object-level shaders. The texture is stored in R16 format, using a single 16-bit channel to represent greyscale density values. This choice reduces memory usage and improves sampling performance, as medical images are inherently greyscale and do not require a full RGB representation. During this stage, the system accesses the previously stored mesh information from the node buffer. Since that data was recorded independently for each object, using local coordinate systems, it is essential that all elements share a common spatial reference. To achieve this, the final shader is rendered on a cubic mesh whose volume is defined by the bounding box of both the segmented 3D model and the medical image volume. Because both datasets originate from the same medical study, this shared volume ensures spatial alignment and avoids distortions during sampling. This setup allows volumetric and mesh-based information to be accurately blended within a unified coordinate framework.

The DVR algorithm is based on the raytracing method, which involves defining a beam that connects a fragment of the mesh to the camera rendering the scene. In this case, the "back-to-front" method is used, where the origin of the ray is the fragment analysed and the camera is considered to be the final position of the ray. This technique allows the best treatment of transparencies. Alternatively, the hybrid rendering algorithm proposed in this paper is also compatible with the "front-to-back" method. In this technique, the ray is evaluated from the scene camera backwards and can be pruned when the accumulated alpha value reaches full opacity.

Once the sample ray has been configured, it is divided into a certain number of segments (or samples), which is a configurable parameter that can be adjusted depending on the performance or quality requirements of the rendering context. All segments of the ray must be equidistant. Along the ray, each segment's data is evaluated and blended with the accumulated data from the previous steps. This is achieved by incorporating alpha blending equations to obtain the final colour of each pixel:

$$C = \sum_{i=1}^{n} C_i \cdot A_i \cdot \prod_{j=1}^{i-1} (1 - A_j) \tag{1}$$

$$A = 1 - \prod_{j=1}^{n} (1 - A_j) \tag{2}$$

where $C$ and $A$ are the accumulated colour and opacity, $C_i$ is the RGB colour of the $i$th sample, and $A_i$ is its corresponding opacity. The product term $\prod_{j=1}^{i-1}(1 - A_j)$ accounts for the transparency of the preceding samples along the ray. This formulation enables a physically plausible accumulation of semi-transparent contributions and provides consistent depth-based compositing in back-to-front ray traversal.

Raymarching techniques allow the analysis of continuous data, such as the density stored in the 3D texture of the medical image. However, the blending of 3D surfaces and the medical image involves the merging of continuous and discontinuous data. Whereas the 3D texture sample contains data for the entire volume, the node buffer contains data sorted by depth. Nevertheless, the data from the node buffer does not need to be uniformly distributed in space. For this reason, the DVR ray sampling system is used as a basis for analysing in which iterations it will be necessary to apply the colour from the 3D models. To ensure consistent blending, the linked list associated with each fragment must be sorted in the same direction as the ray defined for the DVR algorithm. In this way, the regions where the ray intersects a mesh can be detected by comparing the study position of the ray in each sample and the node element in the linked list with the lowest index up to that point. In order to be able to compare the two types of data, it is necessary to define a common space. In our case, the clipping space is chosen because it reflects the depth of each point within a defined range. By using the back-to-front method, the linked list data is only used in the colour blending equation if the depth of each sample's position is less than the depth of the studied linked list node. In the segments where the previous condition is met, the study index of the linked list is increased, traversing the linked list in the same direction as the DVR ray. In this way, all the nodes in the linked list ordered before the current node are ignored because they have already contributed to the colour blending.

A schematic of the data used for the proposed algorithm is shown in Fig. 3. Within the volumetric rendering volume containing the 3D texture from the medical image, there are two 3D meshes (the lungs mesh and the heart mesh). The back-to-front sampling ray is shown in dashed blue and passes through all data types in the scene. However, only the nodes in the ordered linked list that contain data from the 3D meshes are coloured green. This contrast between the continuous nature of the volumetric medical image and the discontinuous structure of the 3D meshes – evidenced by the empty 3D spaces between the green nodes – is a central challenge addressed by the proposed hybrid rendering algorithm.

As shown in Algorithm 2, each step of the ray sampling begins by computing the 3D position at which the sample will be taken. Then, it is evaluated if it is necessary to apply the colour stored in the linked list of the studied fragment before performing the colour blending with the medical image. In this case, the position in the clipping space is calculated both for the DVR sample point and for the world position of the linked list node. With this transformation, both points share a coordinate system and the depth is defined in a limited range of values, making comparisons between the two easier. If the DVR study point has a shallower depth, the additive mixing formula between the accumulated colour and the colour of the node is applied in this case. For the definition of Algorithm 2, we assume a clipping space in which the Far Plane is defined as 1 and the Near Plane's value is 0. Note, however, that the depth range in the clipping space varies depending on the 3D graphics API selected.

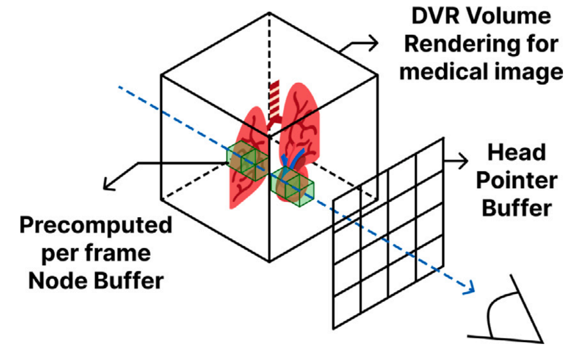



**Fig. 3.** Hybrid rendering scheme. In the foreground is the Head Buffer data structure. The main rendering volume is defined by the box with black borders. The 3D meshes are located inside this volume. The dashed blue arrow represents the back-to-front DVR sampling ray. The green voxels represent only the nodes from the Node Buffer that belong to the linked list of the pixel under study, despite the presence of the full Node Buffer in memory.

**Algorithm 2:** Pseudocode of the final render step: 3D mesh OIT + DVR blending

**Data:** rayStartPos, rayEndPos, rayDirection, NUM_STEPS, HeadBuffer, NodeBuffer, 3DTexture
**Result:** colour C

- $pixelIndex \leftarrow vertex.y \times ScreenParams.x + vertex.x$;
- $pointerRef \leftarrow HeadBuffer[pixelIndex]$ ;
- $LinkedList \leftarrow$ Sort connected nodes in NodeBuffer ;
- $C \leftarrow (0, 0, 0, 0)$ ;
- $nodeIndex \leftarrow 0$;
- **for** $iStep \leftarrow 1$ **to** $NUM\_STEPS$ **do**
  - $currPos \leftarrow$ 3D sampling position along the ray;
  - $curPosClipPos \leftarrow mul(mVP, curPos)$ ;
  - $nodeClipPos \leftarrow mul(mVP, LinkedList[nodeIndex].worldPos)$ ;
  - **if** *(curPosClip.z > nodeClipPos.z) and (geoIndex < LinkedList.length)* **then**
    - $C.rgb \leftarrow C.rgb \times (1 - LinkedList[nodeIndex].C.a) + LinkedList[nodeIndex].C.rgb \times LinkedList[nodeIndex].C.a$;
    - $C.a \leftarrow C.a + LinkedList[geoIndex].C.a$ ;
    - $geoIndex++$;
  - $src \leftarrow sample3D(3DTexture, currPos)$ ;
  - $C.rgb \leftarrow C.rgb \times (1 - src.a) + src.rgb \times src.a$ ;
  - $C.a \leftarrow C.a + src.a$;

To obtain the final colour along the ray, alpha blending techniques are used. As can be seen in Algorithm 2, the same blending equation is applied for both 3D mesh and medical image data, as both types of data have the same relevance for the proposed visualisation system. The influence of the obtained data on the final colour will be defined by the alpha of each node or voxel colour and the depth at which it is located.

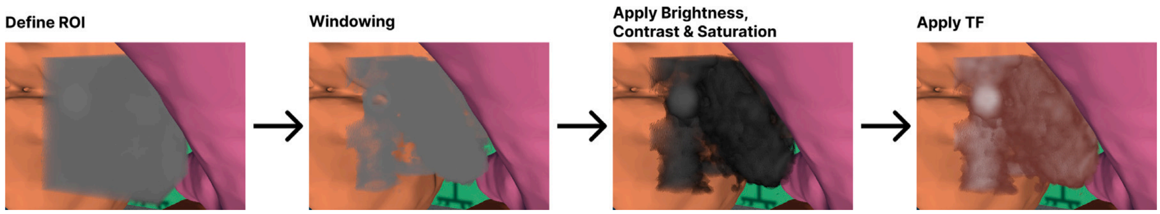



**Fig. 4.** Pipeline for medical image visualisation tools. From left to right, a cube is used as the base geometry to define the ROI. Then, only voxels whose value is within the range determined in the windowing tool are coloured. Next, rendering colour adjustments can be applied to the data from the medical image. Finally, the user can choose whether a colour map is applied or not.

This approach also supports the individual configuration of the alpha values for each element displayed in the scene, enabling fine-grained control over the visibility of structures in the scene.

### 3.2. Additional tools for medical image visualisation

In conjunction with the core hybrid rendering algorithm, a series of additional tools has been implemented to enhance visual analysis and replicate functionalities commonly found in professional medical image viewers. These tools operate in real time and include interactive Region of Interest (ROI), density windowing, global colour adjustments (hue, brightness, contrast, saturation), and transfer function mapping based on voxel density. The results obtained in each step of the pipeline associated with the application of additional tools for medical image visualisation are shown in Fig. 4. A detailed explanation of each tool is presented below.

The ROI tool allows users to define subregions within the rendering volume using basic 3D geometries such as cubes or spheres, with full control over their size, position, and rotation. This enables focused visual inspection and custom blending behaviours within selected areas. To achieve this, a function has been implemented that determines whether the position of the evaluated segment is inside or outside the activated sub-area and defines the final contribution on the blending equation.

The Windowing tool allows users to restrict the displayed range of voxel densities, effectively discarding those outside a specified threshold. This operation is applied directly on the 3D texture, which encodes greyscale voxel values in a single-channel format. Additionally, a dedicated colour adjustment tool enables users to modify the appearance of the medical image by controlling parameters such as hue, brightness, contrast or saturation. These adjustments are applied by temporarily converting the texture data into the HSB colour space and then mapping it back to RGB for the final colour blending stage.

Transfer functions allow density values to be mapped to colour gradients, improving the visual distinction of anatomical structures. A 2D colour ramp defines the palette used for this mapping. All of these operations are integrated into the raymarching pipeline and are applied before the final alpha blending, ensuring consistency with the visual composition model.

Although these tools do not constitute the main technical contribution of this work, they demonstrate the system's applicability in realistic clinical scenarios and its compatibility with immersive and interactive environments.

## 4. Results

The effectiveness of the proposed hybrid rendering algorithm was evaluated across two real-world clinical cases focused on tumour margin assessment. This section summarises the dataset and experimental setup, visual outcomes on different cases, its integration in a VR environment, and performance results on various hardware platforms.

### 4.1. Evaluation scenario and dataset

The main use case for our proposal involves medical specialists evaluating tumour margins in complex surgical cases. In this clinical context, combining volumetric imaging and segmented 3D models is particularly valuable: surface models derived from automatic segmentation may contain inaccuracies, while relying solely on raw medical imaging can be cognitively demanding and time-consuming. The hybrid visualisation approach helps overcome these limitations by merging the clarity of surface models with the raw information of volumetric data.

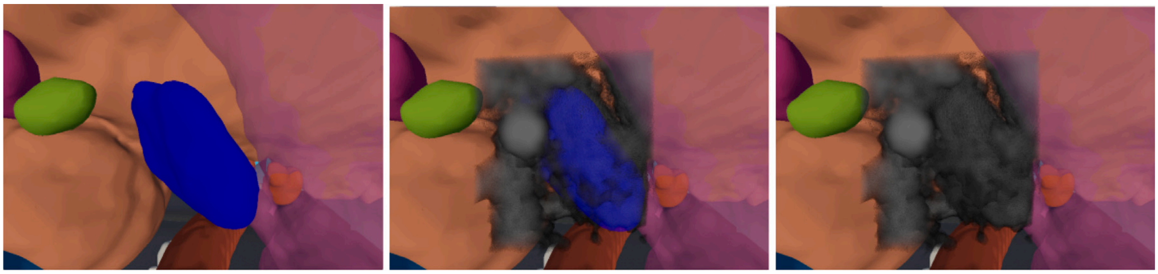
To evaluate the system, we used open-access cancer imaging studies from The Cancer Imaging Archive (TCIA) platform [24]. For each case, a 3D model was generated from the original medical images using 3D Slicer [25] and automatically segmented with the Total Segmentator tool [26]. These datasets provided both volumetric imaging and anatomical models necessary for hybrid visualisation.

### 4.2. Visualisation results across clinical cases

To evaluate the visual output of the proposed algorithm, we rendered two tumour cases using different visualisation configurations. Figs. 5 and 6 illustrate a selection of clinical examples that highlight the contribution of each modality. Each row corresponds to a different tumour case, while each column shows a distinct rendering mode. To replicate a realistic clinical scenario and better visualise internal structures, the ROI tool was activated along with windowing and brightness and contrast adjustments on the medical image.

Fig. 5 shows the tumour mass in blue and the stomach (right side of the view) in pink. The first column suggests that the tumour and stomach are not in contact due to a gap between their geometries, which could be considered a surgical margin. However, the second column shows that the 'empty' area disappears when the medical image is introduced. In this case, a protrusion can be observed from the tumour mass. It is situated halfway up the tumour and fills the space between the tumour mesh and the stomach mesh. This visual configuration applies transparency to the tumour mesh, maintaining the reference of the 3D mesh of the tumour, while the user can see the original data from the medical image. This feature enables easy identification of structures as both tissues in medical images have a similar colour range. The third column displays only the medical image using DVR, which provides a visual representation of the region of interest similar to that obtained with traditional medical imaging inspection tools. However, the use of volumetric rendering and stereoscopy can enhance the sense of depth and aid in identifying volumes in a 3D view.

In addition to the general advantages discussed previously, a key strength of the proposed algorithm lies in its ability to handle scenes with high depth complexity. We understand depth complexity as the number of triangles that has to be depth-based sorted per each pixel. Such configurations are common in real clinical scenarios, particularly when several tissues exhibit complex spatial relationships, including overlap, embedding, or close volumetric proximity. To highlight this feature, a heat map illustrating the per-pixel depth complexity has been incorporated in the results. In this map, colour ranges from dark blue



(a) (b) (c)

**Fig. 5.** Result of applying our hybrid rendering approach to a clinical case. In the examined area, a tumour (blue) is found close to the stomach (pink with transparency). Fig. 5(a) shows the mesh of both tissues. Fig. 5(b) shows the medical image from a CT study mixed with the transparency mesh. In Fig. 5(c), the tumour is visualised with only the medical image. In this case, it is possible to see differences in the volume of the tumour between the 3D mesh from an automatic segmentation process and the representation of the medical image by volumetric rendering. In the right region of the tumour, a protrusion can be seen in the medical image, which increases its volume compared to that represented by the 3D mesh.

(low) to red (high). This visual aid, shown in the rightmost column of Fig. 6, enables visual assessment of each case's complexity and underscores the robustness of the algorithm in accurately blending data.

Fig. 6 presents multiple views of a lung cancer case, highlighting several cancerous nodes. In each row, a specific view is presented for one tumour node, while in each column, the same visualisation mode is used for the different case studies. Analysing in detail the first row of Fig. 6, the visualisation of the medical image volume reveals an irregular volume from which protruding structures can be analysed. In this case, the adjustment of the windowing tool in greater detail is of great complexity since the tissues present in the medical image have very similar attenuation coefficients on the Hounsfield scale [27]. The following image visualises two distinct tissues: the gross tumour volume (GTV), coloured in white, and the set of pulmonary veins and arteries inside the lung cavities. The blending of 3D mesh with volumetric medical imaging facilitates the spatial understanding of the structures present in the image. This innovative hybrid data visualisation mode allows for clear detection of the branching of the pulmonary veins and arteries from the tumour mass, as their volume overlaps the surface of the tumour mass. Nonetheless, the use of 3D meshes should be studied with caution as their surface may contain alterations during the segmentation or mesh generation process. The use of transparencies on 3D meshes facilitates the detection of these incongruities since it allows for visualising in a unified way the depth order of both types of data. An example of this advantage can be seen in the upper central area of the image, where the GTV shows a lump in its upper right corner. This structure is not supported by the underlying medical image data, suggesting a potential artefact introduced during segmentation or mesh generation. To address this, we propose a hybrid and interactive configuration approach that provides the user with access to the entire data and allows for easy configuration of multiple views to determine a prognosis. Despite this complex network of transparencies shown in Depth Complexity column, the algorithm performs an accurate colour mixing of the data from both sources without producing alterations in the order, by transformations in the camera or a drop in the frame rate.

In the second case in the table, the tumour mass is shown in semi-transparent white. The tumour appears to be in contact with the lung wall, which is shown in orange. Applying transparency to the mesh of the tumour mass reveals the difference in volume between the medical image information and the 3D segmented mesh. Of particular note is the elongation observed at the inferior section of the viewport, where the medical image reveals a mass with a morphology and positioning analogous to that delineated by the segmented mesh. This detailed view of the volumes can be crucial for the definition of the resection margin for the tumour and the surgical planning to remove it. Moreover, the application of a high degree of transparency to the lung wall facilitates the observation of the intricate course of the pulmonary circulation structures as they traverse the lung and connect with the vascular system of the thoracic region. On the right side of the image, the structures of the main veins and arteries outside the lung cavity can be seen through the volumes of the medical image.

In the third row, a tumour is highlighted in red to enhance the visibility and clarity of the image, allowing for easier interpretation of the surrounding anatomical structures. In addition, we can visualise the pink coloured middle and lower lobes of the right lung, where the rendering camera is located. The continuous study of the 3D space offered by this hybrid rendering algorithm visualises how the tumour mass intersects both lobes of the lung. In the first image, the transformation applied to the ROI facilitates analysis of the interior of the tumour mass using a vertical section. This functionality offers a view similar to that obtained in the traditional analysis of 2D sections of medical imaging studies without losing the perception of volume and depth offered by volumetric rendering, even without the help of 3D meshes. The second image illustrates the 3D mesh of the tumour, emphasising its total volume. It also facilitates spatial recognition of the structures present in the rendered area by highlighting the percentage of the middle and lower lobe occupied by the tumour. The intersection of 3D meshes is a common source of ordering problems, as the same mesh can appear both in front of and behind a second element. Nevertheless, the proposed method maintains accurate colour blending and stable depth perception, even in scenes with intersecting geometry and high visual complexity.

### *4.3. Immersive visualisation and interaction*

In traditional medical image inspection workflows, clinicians must interpret a large number of 2D slices to mentally reconstruct the 3D anatomy of the tumour and its relation to surrounding tissues. This process can be time-consuming, even with a supporting segmented 3D model, because the medical image contains a lot of information that is difficult to segment in detail.

As an alternative, the proposed immersive virtual reality (VR) application allows the user to configure the visual appearance of each tissue type in the 3D model and use it as a roadmap to localise the area of interest for detailed study with the medical image. Additionally, the medical image can be enhanced in real time using density windowing, transfer functions, and filters for brightness, contrast, hue, and saturation.

The segmented 3D model is initially displayed at a 1:1 anatomical scale. During inspection, the user can freely navigate around the model and adjust its scale. A Region of Interest (ROI) can be activated at any point to focus the analysis. By default, the ROI is defined by the tumour volume, but its shape, size, and position can be interactively modified at any time. The user can switch seamlessly between viewing the 3D model, the medical image alone, or the hybrid rendering of both data sources.

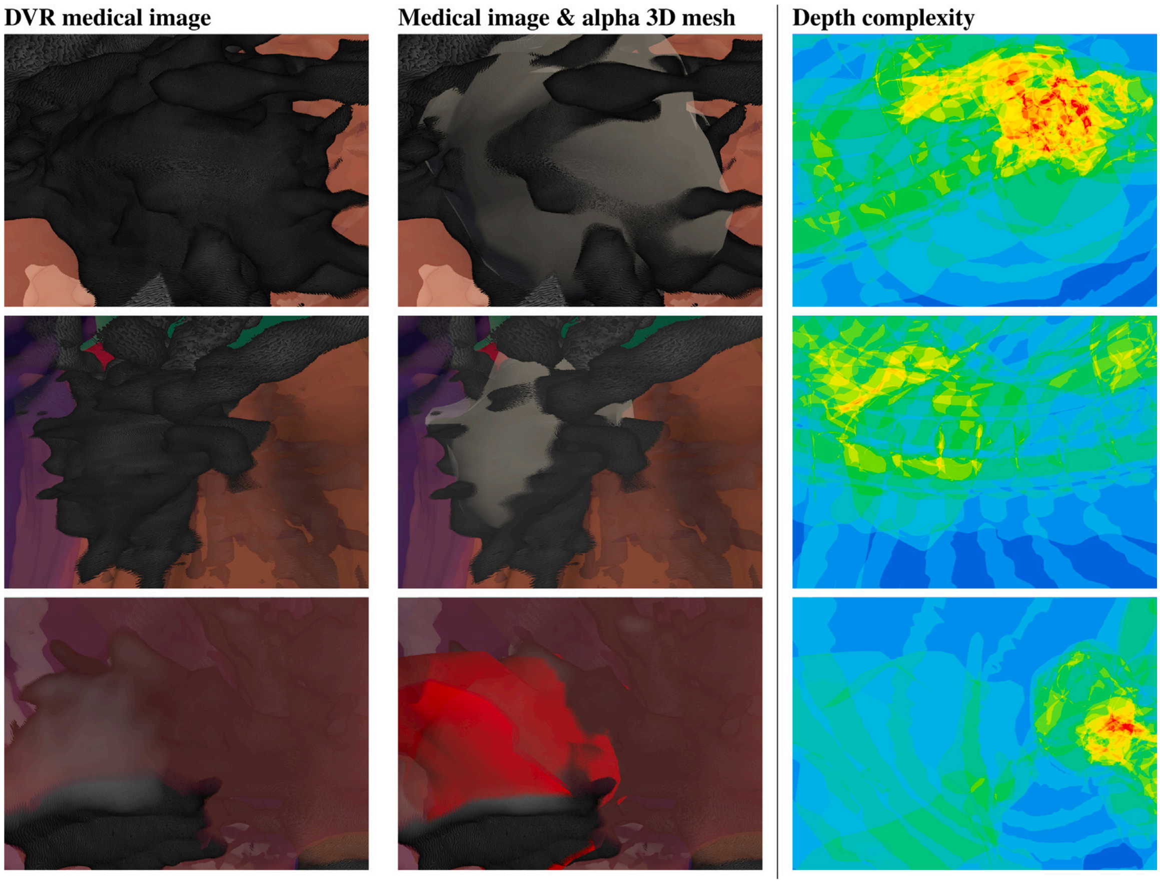



**Fig. 6.** Visualisation results of tumour masses in the thoracic region using different rendering modes. Each row corresponds to a distinct tumour node, and each column represents a specific visualisation configuration: DVR medical image, hybrid transparent mesh with DVR, and depth heat map. The hybrid approach enables the detection of segmentation errors, protrusions not captured in the mesh, and complex spatial relationships between tumour masses and surrounding vasculature structures.

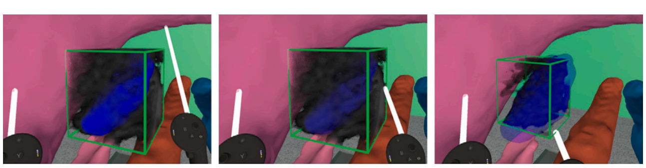


**Fig. 7.** Sequence of user interaction with the virtual environment using the VR HMD & controllers. The user is free to move the ROI box around the blue tumour and change its size. This visualisation tool helps users to compare 3D mesh shapes and medical image volumes.

All tools are mapped to the buttons and joysticks of the VR Head-Mounted Display (HMD) controllers. This allows users to perform the entire visual configuration and inspection process without diverting attention from the ROI and observe how each interaction affects the final blended rendering in real time. An undo option is also provided to revert to previous states. For mesh-specific adjustments, the user can employ the laser pointer of the HMD controller to disable individual meshes or modify their transparency level, enabling precise interaction with elements in the scene. Fig. 7 shows an example of this interaction, in which the user adjusts the transparency of the central tumour and modifies the position and size of the ROI to obtain a suitable view for clinical analysis.

This immersive configuration complements the hybrid rendering approach by providing enhanced spatial perception, intuitive interaction, and a more natural integration of the visualisation tools explored in the previous sections.

### 4.4. Performance

To assess the performance of the proposed rendering algorithm under realistic conditions, the two clinical cases shown in Figs. 5 and 6 were evaluated, replicating the visualisation settings used in those examples. Each case includes a CT image volume obtained from the TCIA database [24], along with a segmented 3D model generated using the Total Segmentator tool [26]. Performance tests focused on hybrid rendering of these two data sources – volumetric medical imaging and its associated 3D model – under typical inspection scenarios.

**Case 1**, shown in Fig. 5, corresponds to a lymphoma located near the lung. The dataset includes 661 slices at a resolution of $512 \times 512$ pixels, with study volume dimensions of $38.22 \times 38.17 \times 65.94$ cm. The size of the medical image dataset is 0.65 GB, stored in a single-channel format for efficient volume usage. The segmented 3D model derived from this study contains 1 013 501 vertices. In the visualisation shown, the rendered scene includes approximately 137 800 vertices and 263 000 triangles.

**Case 2**, illustrated in Fig. 6, corresponds to a lung cancer case based on a CT study comprising 178 slices, each with a resolution of $512 \times 512$ pixels, and study volume dimensions of $49 \times 49 \times 89$ cm. In this case, the size of the medical image data is 178 MB. The segmented 3D model for this case contains 261 990 vertices and 523 507 triangles. The rendered view includes approximately 213 800 vertices and 411 200 triangles.

To establish a baseline for performance comparison, the C++ implementation of the VTK library was used to configure a hybrid visualisation pipeline combining direct volume rendering with the rendering

**Table 1**
Hardware configurations used for performance testing.

| System | Processor | GPU |
|---|---|---|
| PC 1 | Intel Core i7-7700HQ | NVIDIA GeForce GTX 1050 Ti |
| PC 2 | Intel Core i7-10870H | NVIDIA GeForce GTX 1650 |
| PC 3 | Intel Core i5-12400F | NVIDIA GeForce GTX 1060 |
| PC 4 | Intel Core i7-8700K | AMD Radeon VII |
| PC 5 | AMD Ryzen 5 3600 | NVIDIA GeForce RTX 3060 |
| PC 6 | Intel Core i9-14900KF | AMD Radeon RX 6700 XT |

of semi-transparent 3D segmented model using the depth peeling technique. The evaluation was conducted using PC 4, described in Table 1, and the dataset from Case 2, using camera angles and transparency settings equivalent to those in our system. To ensure correct visual layering and minimise ordering artefacts, the depth peeling configuration was set to allow up to 20 depth layers per pixel. The viewport defined for this implementation was 1280 × 720 pixels.

Rendering timing measurements were collected over 100 iterations by capturing timestamps immediately before and after the rendering pipeline execution. The measured average render time per frame was 59.5 milliseconds (ms), with a standard deviation of 6.3 ms. This corresponds to an effective framerate of approximately 16.8 frames per second, which is significantly below the threshold required for real-time interaction or immersive exploration.

Given this result, the VTK-based configuration was considered unsuitable for responsive hybrid rendering, especially in contexts requiring fluid user interaction with both volumetric and geometric data. This motivated the design and evaluation of the GPU-based algorithm presented in this paper.

Subsequent performance evaluation focused on the proposed algorithm, tested on seven desktop systems equipped with mid-range and high-end GPUs. Integrated graphics were excluded, given the application's high computational demands, particularly for high-fidelity rendering of complex volumetric and geometric content, which imposes minimum requirements in terms of GPU memory. Maintaining visual accuracy without relying on data reduction techniques is essential, especially in contexts involving clinical interpretation or detailed spatial inspection. Table 1 summarises the hardware specifications of the systems used for benchmarking and performance validation.

Performance measurements were carried out in desktop mode at a fixed resolution of 1280 × 720 pixels, using 1024 sampling steps. This configuration was selected to achieve high-quality results, allowing for detailed colour blending across semi-transparent regions and precise layering of volumetric structures. To ensure consistent and reproducible conditions, the render camera followed a predefined motion path across all tests. This setup reflects interactive usage scenarios while providing a controlled benchmark environment. The system was evaluated in real time, capturing the render time of the first 100 frames in each session. For every frame, individual time spans were recorded to compute the average frame duration and its standard deviation, thereby assessing both runtime efficiency and temporal stability. Although immersive visualisation is a key application of the proposed method, benchmarking was limited to desktop mode, as replicating identical rendering conditions in VR is inherently difficult—the user controls the viewing direction dynamically, making frame-by-frame comparisons impractical. The system has been functionally tested in VR and supports real-time interaction in that context.

Table 2 presents the measured rendering performance for each system, expressed as the average frame duration and standard deviation over 100 consecutive frames. The results are given in milliseconds and confirm that the proposed hybrid algorithm delivers interactive performance across a wide range of GPU configurations. All systems tested achieved average frame times well below 1 ms, except for PC 1 (1.8 ms), which is a laptop with low-end hardware. Low standard deviation across systems reflects high temporal consistency and stability.

**Table 2**
Measured rendering performance for each clinical case. Values represent average frame duration (in milliseconds) and standard deviation over 100 frames. Number of sampling: 1024.

| System | Case 1 | | Case 2 | |
|---|---|---|---|---|
| | Avg | Std Dev | Avg | Std Dev |
| PC 1 | 1.8435 | 0.4452 | 1.3385 | 0.3787 |
| PC 2 | 0.7624 | 0.1674 | 0.5149 | 0.0539 |
| PC 3 | 0.6532 | 0.0755 | 0.5563 | 0.0728 |
| PC 4 | 0.8322 | 0.0703 | 0.6455 | 0.0722 |
| PC 5 | 0.4747 | 0.0519 | 0.3291 | 0.0225 |
| PC 6 | 0.2605 | 0.0347 | 0.2329 | 0.0473 |

**Table 3**
Experts evaluation questionnaire.

| Question | Question text |
|---|---|
| Q1 | I believe the presented visualisation allows me to understand the depth ordering of the semi-transparent 3D elements. |
| Q2 | I believe the volumetric visualisation of the medical image enables me to infer complex structures or elements not available in the segmented 3D models. |
| Q3 | I believe that visualising the 3D tissues alongside the medical image enables a more accurate understanding of the clinical information. |
| Q4 | I believe that the use of this rendering algorithm could enhance the analysis of a clinical case prior to surgical intervention. |
| Q5 | I believe the hybrid rendering algorithm is useful for integrating the various types of information associated with a clinical case. |
| Q6 | I believe this rendering algorithm facilitates the interpretation of medical imaging compared to traditional methods. |

Even on less powerful hardware (e.g., PC 1 and PC 2), the algorithm sustained fluid responsiveness, validating its suitability for real-time exploration of volumetric and surface data. Low standard deviations further demonstrate that performance remains robust under continuous interaction, without perceptible spikes or frame drops. Explicit memory benchmarking was not performed, as it has been observed that memory usage remains constant across different computers. The rendering algorithm allocates around 0.65 GB of GPU memory for Case 1 and 0.79 GB for Case 2, containing textures and data structures.

### 4.5. Expert evaluation

To assess the clinical relevance and perceived utility of the proposed hybrid rendering system, we conducted a structured evaluation with nine healthcare professionals specialised in robotic-assisted surgery (RAS). This specific clinical profile was intentionally selected given its direct involvement in preoperative planning and intraoperative decision-making, where spatial understanding of anatomical relationships is critical. The evaluation began with an introduction to the clinical case used in this study, including detailed information about the patient's pathology. Participants then were presented with the different visualisation modes applied to that same case, and subsequently completed a 5-point Likert scale (1: strongly disagree, 5: strongly agree) questionnaire evaluating the system's usability and clinical applicability. Table 3 lists the questions of the experts questionnaire.

Fig. 8 presents the outcomes of the expert evaluation. The first three questions (**Q1–Q3**) addressed the participants' perception of spatial accuracy and information integration. Regarding **Q1**, most respondents agreed that the hybrid rendering facilitated understanding of the depth ordering among semi-transparent 3D structures. **Q2** focused on the volumetric visualisation of the medical image, with a clear majority acknowledging its effectiveness in revealing complex anatomical details that may be absent in segmented models.

In **Q3**, experts evaluated whether combining 3D models and volumetric imaging improved comprehension of the clinical case. Half of

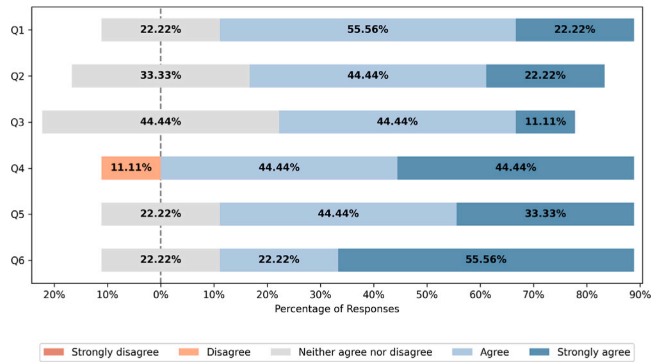



**Fig. 8.** Experts' responses to the five-point Likert-scale questionnaire used to evaluate spatial perception, integration of visual information and clinical utility.

the participants agreed, while the remaining responses were neutral—none expressed disagreement. This indicates a generally positive reception, albeit potentially influenced by the respondents' varying familiarity with immersive 3D visualisation, which continues to evolve in clinical practice.

The last three items (**Q4–Q6**) focused on the clinical applicability of the algorithm. For **Q4**, most participants considered the hybrid rendering approach beneficial for preoperative analysis, with only one expressing disagreement. **Q5** further confirmed this positive trend, showing that the majority recognised the system's potential to unify and present case-related information in an integrated manner. **Q6** explored whether the hybrid rendering approach improves the interpretation of medical imaging when compared to traditional visualisation techniques. Most participants responded positively, suggesting that the method enhances interpretability and supports clinical decision-making.

Qualitative feedback reinforced these observations. One participant noted its direct "usefulness for surgical planning", while another emphasised its "simplicity and ability to quantify lesion dimensions and distances to access paths or vital structures". Together, the quantitative and qualitative responses suggest that the hybrid rendering system supports a more informed and spatially grounded understanding of complex clinical cases. These findings confirm both the technical soundness and the clinical relevance of the proposed method, particularly in surgical contexts that demand precise spatial reasoning and comprehensive data integration.

## 5. Conclusions and future work

We present a rendering algorithm that enables hybrid visualisation of semi-transparent 3D meshes and volumetric medical imaging. The proposed technique combines the advantages of seamless transparency blending with high-quality volumetric rendering, facilitating the interpretation of structures that are typically difficult to assess in traditional medical image inspection. One key advantage of this combination is that it maintains the intuitive and direct interaction capabilities of segmented 3D models – such as spatial navigation, annotations, or tissue-specific exploration – while enriching the context with raw volumetric information that may not be captured through segmentation alone. The algorithm is lightweight and suitable for integration into automated clinical pipelines, allowing specialists to explore cases using both data types without requiring manual data preparation by additional personnel.

The method was evaluated by nine specialists in robotic-assisted surgery using a structured questionnaire. The results showed that most participants perceived improved depth understanding and recognised the volumetric view as helpful in revealing anatomical details not present in segmented models. The majority considered the hybrid rendering useful for preoperative analysis and case review. Qualitative feedback highlighted its simplicity and its value for surgical planning.

The methods outlined in this paper serve as fundamental elements for crafting tailored rendering approaches suited to specific visualisation needs within various applications. We believe that this approach represents a promising direction for the future of scientific visualisation. Successful techniques should be customised to address specific visualisation challenges, blending the technical proficiency of computer scientists with the domain-specific insights of clients to develop solutions that are user-friendly for both parties involved. Our proposed algorithm allows the visualisation of continuous and discontinuous datasets in an efficient and precise way. One of its key points is to use a system optimised for continuous data analysis to perform the blending with the discontinuous data set. To achieve this, it was necessary to apply transformations to the data obtained to create a common space in which they could be accurately evaluated.

Despite its strengths, the method has limitations that warrant further investigation. First, whereas the algorithm achieves real-time performance on standard datasets, highly complex scenes with deep depth layering may lead to node buffer overflows. A potential improvement involves dynamically adjusting the size of the node buffer according to the number of depth layers encountered in each frame, ensuring better memory management and avoiding overflow without over-allocating resources. Second, although the system has been validated on CT data, further studies are needed to confirm its adaptability to other imaging modalities such as MRI or PET.

The hybrid rendering technique was developed to analyse complex cases in the healthcare sector and reduce surgical specialists' reliance on radiologists' reports. Additional visualisation tools have been proposed to support this approach. Furthermore, incorporating quantitative analysis tools directly into the VR environment could provide clinicians with real-time metrics related to tumour volume, tissue density, and spatial relationships, enhancing the practical utility of the system. Overlapping 3D mesh volumes with a medical image can be an interesting research line in order to automatically calculate the windowing parameters for enhancing the tumour mass in the medical image volume rendering view. Additionally, conducting extensive usability studies with medical professionals would offer valuable insights for refining the system's interface and interaction methods, ensuring that it meets clinical requirements and user expectations. We aim to establish our hybrid rendering technique as a standard tool for advanced medical visualisation by continuously evolving the algorithm based on clinical feedback and technological advancements.

Finally, future work includes testing the proposed algorithm in other domains where complex data blending is required, expanding its applicability beyond the healthcare sector. Potential applications include scientific visualisation, engineering simulations, and any scenario that demands the integration of heterogeneous spatial data. The method's flexibility and real-time performance make it a promising candidate for domain-adapted rendering pipelines in various technical fields.

## CRediT authorship contribution statement

**Belén Palma:** Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Formal analysis. **Pablo Casanova-Salas:** Writing – review & editing, Validation, Investigation. **Jesús Gimeno:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis. **Sergio Casas-Yrurzum:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Formal analysis, Data curation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Belén Palma, Jesús Gimeno and Sergio Casas are supported by Spain's Agencia Estatal de Investigación and the Spanish Ministry of Science and Innovation, under the ViPRAS project (Virtual Planning for Robotic-Assisted Surgery)-PID2020-114562RA-I00 (funded by MCIN/AEI/10.13039/501100011033).

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2025.104453.

## Data availability

Data will be made available on request.

## References

[1] Wake N, Wysock JS, Bjurlin MA, Chandarana H, Huang WC. "Pin the tumor on the kidney":An evaluation of how surgeons translate CT and MRI data to 3D models. Urology 2019;131:255–61. http://dx.doi.org/10.1016/j.urology.2019.06.016.

[2] Abhari K, Baxter JSH, Chen ECS, Khan AR, Peters TM, de Ribaupierre S, Eagleson R. Training for planning tumour resection: Augmented reality and human factors. IEEE Trans Biomed Eng 2015-06;62(6):1466–77. http://dx.doi.org/10.1109/TBME.2014.2385874.

[3] Necker FN, Melcher ML, Busque S, Leuze CW, Ghanouni P, Le Castillo C, Nguyen E, Daniel BL. Nested Semi-Transparent Isosurface Simulated Volume-Rendering (NESTIS-VR) – an efficient on-device rendering approach for augmented reality headsets increasing surgeon confidence of kidney donor arterial anatomy. Comput Biol Med 2024;183:109267. http://dx.doi.org/10.1016/j.compbiomed.2024.109267.

[4] Liu X, Song L, Liu S, Zhang Y. A review of deep-learning-based medical image segmentation methods. Sustainability 2021-01;13(3):1224. http://dx.doi.org/10.3390/su13031224.

[5] Fick T, Meulstee JW, Köllen MH, Van Doormaal JAM, Van Doormaal TPC, Hoving EW. Comparing the influence of mixed reality, a 3D viewer, and MRI on the spatial understanding of brain tumours. Front Virtual Real 2023;4. http://dx.doi.org/10.3389/frvir.2023.1214520.

[6] Bavoil L, Myers K. Order independent transparency with dual depth peeling. NVIDIA OpenGL SDK 2008;1:12.

[7] McGuire M, Bavoil L. Weighted blended order-independent transparency. J Comput Graph Tech 2013;2(4).

[8] Barta P, Kovács B, Szecsi SL, Szirmay-kalos L. Order independent transparency with per-pixel linked lists. Bp Univ Technol Econ 2011;3.

[9] Drebin RA, Carpenter L, Hanrahan P. Volume rendering. ACM SIGGRAPH Comput Graph 1988;22(4):65–74. http://dx.doi.org/10.1145/378456.378484.

[10] Xu C, Sun G, Liang R. A survey of volume visualization techniques for feature enhancement. Vis Inform. 2021;5(3):70–81. http://dx.doi.org/10.1016/j.visinf.2021.08.001.

[11] Praßni J-S, Ropinski T, Mensmann J, Hinrichs K. Shape-based transfer functions for volume visualization. In: 2010 IEEE Pacific visualization symposium. Pacificvis, 2010-03, p. 9–16. http://dx.doi.org/10.1109/PACIFICVIS.2010.5429624.

[12] Kniss J, Kindlmann G, Hansen C. Multidimensional transfer functions for interactive volume rendering. IEEE Trans Vis Comput Graph 2002-07;8(3):270–85. http://dx.doi.org/10.1109/TVCG.2002.1021579.

[13] Maciejewski R, Woo I, Chen W, Ebert D. Structuring feature space: A non-parametric method for volumetric transfer function generation. IEEE Trans Vis Comput Graph 2009-11;15(6):1473–80. http://dx.doi.org/10.1109/TVCG.2009.185.

[14] Ma J, Chen J, Chen L, Zhou X, Qin X, Tang Y, Sun G, Chen J. Gaussian mixture model-based target feature extraction and visualization. J Vis 2021;24(3):545–63. http://dx.doi.org/10.1007/s12650-020-00724-0.

[15] Agus M, Gobbetti E, Guitián JAI, Marton F. Split-voxel: a simple discontinuity-preserving voxel representation for volume rendering. In: Proceedings of the 8th IEEE/EG international conference on volume graphics. 2010, p. 21–8.

[16] Schroeder W, Martin K, Lorensen B. The Visualization Toolkit. 4th ed.. Kitware; 2006.

[17] Rezk-Salama C, Kolb A. Opacity peeling for direct volume rendering. Comput Graph Forum 2006;25(3). http://dx.doi.org/10.1111/j.1467-8659.2006.00979.x.

[18] Chu Y, Li X, Yang X, Ai D, Huang Y, Song H, Jiang Y, Wang Y, Chen X, Yang J. Perception enhancement using importance-driven hybrid rendering for augmented reality based endoscopic surgical navigation. Biomed Opt Express 2018;9(11):5205–26. http://dx.doi.org/10.1364/BOE.9.005205.

[19] Merten N, Glaßer S, Lassen-Schmidt B, Großer OS, Ricke J, Amthauer H, Preim B. Illustrative PET/CT visualisation of SIRT-treated lung metastases. In: Proceedings of the eurographics workshop on visual computing for biology and medicine. VCBM '16, Goslar, DEU: Eurographics Association; 2016, p. 99–103. http://dx.doi.org/10.2312/vcbm.20161276.

[20] Patel D, Šoltészová V, Nordbotten JM, Bruckner S. Instant convolution shadows for volumetric detail mapping. ACM Trans Graph 2013;32(5). http://dx.doi.org/10.1145/2492684.

[21] Yang JC, Hensley J, Grün H, Thibieroz N. Real-time concurrent linked list construction on the GPU. Comput Graph Forum 2010;29(4):1297–304. http://dx.doi.org/10.1111/j.1467-8659.2010.01725.x.

[22] Frick C, Rumgay H, Vignat J, Ginsburg O, Nolte E, Bray F, Soerjomataram I. Quantitative estimates of preventable and treatable deaths from 36 cancers worldwide: A population-based study. Lancet Glob Heal 2023-11-01;11(11):e1700–12. http://dx.doi.org/10.1016/S2214-109X(23)00406-0.

[23] Palma B, Portalés C, Coma I, Gimeno J, Casas-Yrurzum S. On the design and implementation of a mixed-platform collaborative virtual RAS planner. Appl Sci 2024;14(8):3349. http://dx.doi.org/10.3390/app14083349.

[24] Prior FW, Clark K, Commean P, Freymann J, Jaffe C, Kirby J, Moore S, Smith K, Tarbox L, Vendt B, Marquez G. TCIA: An information resource to enable open science. In: 2013 35th annual international conference of the IEEE engineering in medicine and biology society. EMBC, 2013, p. 1282–5. http://dx.doi.org/10.1109/EMBC.2013.6609742.

[25] Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin J-C, Pujol S, Bauer C, Jennings D, Fennessy F, Sonka M, Buatti J, Aylward S, Miller JV, Pieper S, Kikinis R. 3D Slicer as an image computing platform for the quantitative imaging network. Magn Reson Imaging 2012;30(9):1323–41. http://dx.doi.org/10.1016/j.mri.2012.05.001.

[26] Wasserthal J, Breit H-C, Meyer MT, Pradella M, Hinck D, Sauter AW, Heye T, Boll D, Cyriac J, Yang S, Bach M, Segeroth M. TotalSegmentator: robust segmentation of 104 anatomical structures in CT images. Radiol Artif Intell 2023;5(5):e230024. http://dx.doi.org/10.1148/ryai.230024.

[27] DenOtter TD, Schubert J. Hounsfield Unit. In: statPearls[internet]. Treasure Island (FL): StatPearls Publishing; 2023, URL: http://www.ncbi.nlm.nih.gov/books/NBK547721/.