doi:10.3969/j.issn.1005-3697.2026.04.002

❖智能医学研究专题❖

Implementation of an interactive chest CT image demonstration system based on unreal engine

HE Wen-jing[1], LI Bing[2], TAN Xue-feng[1], LIAO Qin-mei[1], ZENG Zhen[3], TAN Shi-han[4], LI Xin[5], CHEN Xiao-wen[1]

(1. *School of Medical Imaging, North Sichuan Medical College*; 2. *Affiliated Hospital of North Sichuan Medical College, Nanchong* 637000; 3. *Chengdu University of Information Technology*; 4. *Wisesoft Co., Ltd., Chengdu* 610045; 5. *Nanchong Electronic Information Industry Technology Research Institute, Nanchong* 637000, *Sichuan, China*)

【Abstract】 Objective: Traditional medical image visualization technologies present several limitations including the poor intuitiveness of 2D images, overlapping imaging blind spots, and insufficient interactivity. These shortcomings make them unable to satisfy the demands of precision medicine and medical education. This study aims to develop a universal medical image demonstration system based on extended reality (XR) technology, which is compatible with devices such as the HoloLens. By employing 3D visualization and multimodal interaction design, the system provides a more efficient and intuitive approach for clinical diagnosis, medical education, and surgical simulation. **Methods**: The system employs Unreal Engine as the core platform for architecture framework construction, supporting customized medical image visualization for XR devices. Real chest CT images were utilized in this study. Image segmentation was performed using 3D Slicer, while mask preprocessing was conducted via Anaconda. The masked images and raw data were then imported into Unreal Engine. The core framework of the proposed system was developed using Unreal Engine's Blueprint visual scripting and Unreal Motion Graphics (UMG) interface designer. A self-directed learning assessment experiment was designed to evaluate the efficacy and performance of the system. **Results**: The experimental group achieved a pulmonary window recognition accuracy of 82%, a mediastinal window accuracy of 88%, and an overall accuracy of 85%. By comparison, the control group exhibited a pulmonary window accuracy of 50%, a mediastinal window accuracy of 44%, and an overall accuracy of 47%. **Conclusions**: The medically oriented imaging demonstration system significantly improves recognition accuracy and learning effectiveness, verifying its practical utility and validity.

【Key words】 Medical image visualization; Chest CT; Unreal engine

1 Introduction

Chest computed tomography (CT) is an imaging modality that employs X-ray beams to acquire multi-planar scans of the human thorax. It provides high-resolution cross-sectional images and enables the reconstruction of coronal, sagittal and three-dimensional views via post-processing techniques [1]. This technology clearly delineates the anatomical details of the lung parenchyma, tracheobronchial tree, mediastinal structures (including the heart, great vessels, esophagus, and lymph nodes), pleura, chest wall, and bony structures. It exhibits significantly superior detection performance for small lesions compared with conventional chest radiography [2]. In clinical practice, chest CT plays an irreplaceable role in the diagnosis and assessment of a wide range of diseases. These include pulmonary disorders such as pneumonia [3-4], pulmonary tuberculosis [5-6], pulmonary embolism, emphysema[7], bronchiectasis [8], and other pulmonary disorders, as well as mediastinal tumors, thoracic aortic aneurysms, and pleural mesothelioma[9-11]. Additionally, computed tomography angiography (CTA) can clearly visualize the pulmonary arteries, aorta, and their respective branches, thereby facilitating the diagnosis of life-threatening conditions such as pulmonary embolism[12] and aortic dissection[13].

As a powerful real-time 3D creation platform, Unreal Engine (UE) integrates advanced core technologies including dynamic global illumination (Lumen) and virtualized geometry (Nanite) enabling cinematic-quality, high-fidelity real-time

Corresponding anthor: CHEN Xiao-wen. E-mail: 56833804@qq.com

rendering and efficient cross-platform deployment across PCs, XR devices, and other hardware. Owing to these advantages, UE has become a preferred platform for developing healthcare visualization, digital twin, and mixed reality (MR) applications, extending its application scope well beyond traditional gaming and film industries[14-16].

In the domain of medical imaging education and clinical demonstration, current systems still present a series of critical challenges. Although traditional 2D imaging modalities (e. g., X-ray and axial CT images) exhibit high computational efficiency, their inherent projective overlap results in the occlusion of anatomical structures. This not only reduces diagnostic accuracy but also substantially increase the learning burden for students[17]. To address the limitations of 2D display, immersive technologies including virtual reality, augmented reality, and mixed reality (VR/AR/MR) have gained increasing attention[18-20]. These technologies deliver immersive and interactive learning experiences while conveying precise spatial 3D information, showing considerable potential in nursing skill training, medical education, and cognitive-affective[21-22]. However, existing systems are often developed using low-level graphics libraries, leading to high development complexity and limited functional scalability. Moreover, their implementation faces substantial challenges in resource-limited settings, particularly in developing countries, which hinders widespread clinical adoption[23]. More critically, current systems generally lack robust multi-modal data fusion capability, and cannot effectively integrate multi-source imaging data such as CT, MRI, and ultrasound. This deficiency restricts their performance and practical value thereby limiting their effectiveness in clinical decision-making and medical education.

To address the aforementioned challenges, this study innovatively employs Unreal Engine as an integrated development framework. We hypothesize that a system constructed on this engine can outperform conventional methods and existing XR systems by enabling more accurate and efficient multi-modal image fusion alongside high- fidelity real-time rendering. Furthermore, it strives to deliver a more stable and intuitive interactive experience across PC and VR platforms, ultimately offering a more efficient and user-friendly solution for clinical diagnosis, medical education and training, and surgical planning.

2 Methods

2.1 Data Acquisition and Processing

The CT data utilized to validate the proposed method in this study were obtained from the Department of Radiology, the Affiliated Hospital of North Sichuan Medical College, with prior ethical approval. All data were fully anonymized prior to processing and analysis. To guarantee data quality and representativeness, a single case was randomly selected from the archived dataset according to the following criteria. The inclusion criteria were: (1) chest CT images from a healthy adult male, (2) complete anatomical coverage during scanning, and (3) artifact-free images with satisfactory clarity. The exclusion criteria were: (1) evidence of thoracic pathologies and (2) suboptimal image quality. The final dataset used in this study comprised one DICOM-format case that met all the aforementioned criteria.

To implement the hover-based anatomical structure identification function, manual segmentation was first performed on the original CT images to generate region-specific label maps for distinguishing different tissue types. Segmentation was conducted by a radiologist with 6 years of clinical experience, who imported the raw DICOM data into 3D Slicer (Version 5. 4. 0, www. slicer. org) and performed two sets of manual annotations: (1) Lung window annotations: 4 labeled structures (trachea, lungs, heart, intestines); (2) Mediastinal window annotations: 8 labeled structures (trachea, lungs, heart, intestines, spinal cord, liver, spleen, kidneys). After importing the data into 3D Slicer, a semi-automated segmentation strategy was adopted. For elongated anatomical structures with minimal inter-slice variation (e. g., trachea, lungs, and spinal cord), manual segmentation was performed on every alternate axial slice using Paint and Draw tools. The "Fill Between Slices" algorithm was subsequently applied to automatically interpolate and generate

labels for the intermediate slices. For the remaining anatomical structures, slice-by-slice manual segmentation was conducted to ensure segmentation accuracy.

After data processing, it was necessary to acquire original lung window images, original mediastinal window images, and their corresponding mask images that are compatible with UE.

The original image processing was performed as follows: The initial and final slices were determined according to the thoracic region labels in 3D Slicer and stored in NRRD format. Using Python's SimpleITK library in Python, all slices between the initial and final slices were extracted from the original images data. These slices were then converted to PNG format (512×512 pixels) via the matplotlib library with the corresponding window settings: 1500-2000 HU (width) and -500 HU (level) for lung window, and 250-350 HU (width) with 40 HU (level) for mediastinal window, as shown in Fig. 1.

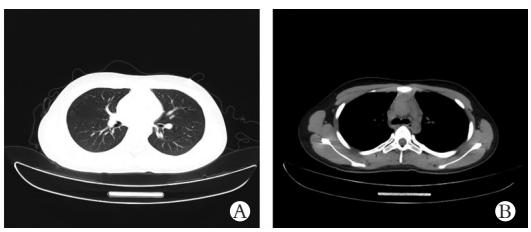

Fig.1 PNG-format images

A. Original pulmonary window images; B. Original mediastinal window images.

The mask image processing was performed as follows. Two sets of mask images were exported from 3D Slicer. Following format conversion, the grayscale value of the two mask sets were distributed as {255, 254, 253, 252, 0} and {255, 254, 253, 252, 251, 250, 249, 248, 0}, respectively. It was observed that non-labeled regions corresponded to a grayscale value of 0, whereas labeled regions exhibited grayscale values clustered near 255. In UE, pixel values approaching 255 may undergo clamping during recognition. Furthermore, the excessively dense distribution of grayscale values in the mask images increased the risk of clamping. Accordingly, grayscale mapping was implemented for subsequent image processing.

In this study, the eight non-zero grayscale values from both sets of mask images were linearly mapped to 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, and 0.3 times 255, respectively. This mapping scheme preserved an interval of approximately 25 between adjacent values, ensuring sufficient discriminability between different organ regions. The mask images following grayscale mapping are presented in Fig. 2.

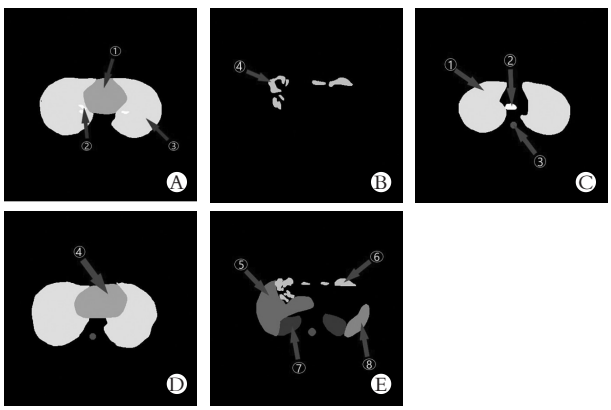

Fig.2 Mask images after grayscale mapping.

A–E. show pulmonary window images, where ① indicates the heart, ② the trachea, ③ the lungs, and ④ the intestines. C–E. display mediastinal window images, where ① represents the lungs, ② the trachea, ③ the spinal cord, ④ the heart, ⑤ the liver, ⑥ the intestines, ⑦ the kidneys, and ⑧ the spleen.

Following extraction of the label grayscale information, the data were mapped to RGBA images via channel duplication, where "A" = 0 denotes full transparency (100% transparency). An information table designated DT_OrganData was constructed to store tissue-specific parameters. Partial contents of DT_OrganData are summarized in Table 1, including the index number, anatomical name, corresponding RGBA values, and medical description.

2.2 System Construction

Unreal Motion Graphics (UMG) functions as the dedicated user interface (UI) development framework for UE. It uses control blueprints to construct interactive interfaces including Head-Up Displays (HUDs) and menu systems. The designer module supports the visual layout of interface widgets such as buttons and progress bars while the graph editor is used to implement interaction logic and data binding. The system features natively supports multi-resolution adaptation and dynamic layout adjustment, and enables sophisticated visual effects through style configuration and the animation subsystem. UMG adopts a three-

layer architecture consisting of a C++ underlying layer, a blueprint intermediate layer, and a plugin extension layer. This design achieves a favorable balance between flexibility and development efficiency, establishing UMG as a core solution for professional UI development in industry applications.

Blueprint visual scripting constitutes one of the UE's core functionalities. Nodes serve as the fundamental building blocks of the Blueprint visual scripting system, enabling the graphical implementation of game logic and interactive functions. Based on their primary functions, these nodes can be categorized into five types: event-response nodes, function nodes, flow-control nodes, data-operation nodes, and UI nodes.

Table 1 Information contained in the DT_OrganData dataset

Row Name	Organ Name	Color ID	Text
New Row	Trachea	255.255.255.0	The trachea is a tubular structure of the respiratory system that connects superiorly to the larynx and inferiorly bifurcates into the left and right main bronchi.
New Row_0	Lungs	229.229.229.0	The trachea bifurcates into the main bronchi, which deliver air to the lungs—paired thoracic organs located superior to the heart
New Row_1	Intestine	204.204.204.0	The intestine extends from the gastric pylorus to the anus, and serves as the longest and functionally most important segment of the digestive tract
New Row_2	Heart	178.178.178.0	The heart is located in the middle-left portion of the thoracic cavity, and its primary function is to drive systemic blood circulation
New Row_3	Spleen	153.153.153.0	The spleen is a essential lymphoid organ situated in the left upper abdomen, presenting as a flattened ovoid structure with a dark red appearance dark red appearance, soft texture, and fragile consistency
New Row_4	Liver	127.127.127.0	The liver is a vital organ responsible for regulating most metabolic processes in the human body
New Row_5	Spinal Cord	102.102.102.0	The spinal cord is a cylindrical structure, flattened antero-posteriorly and exhibiting regional variations in diameter along its course. It resides within the vertebral canal, extending from the foramen magnum (where it is continuous with the medulla oblongata) caudally to a conical distal termination
New Row_6	Kidneys	76.76.76.0	The kidneys are paired bean-shaped organs located in the retroperitoneal space, nestled within shallow fossae on either side of the vertebral column

2.2.1 Implementation of the MainSurface UI Interface

The UI comprises two distinct components: (1) the visual interactive interface elements that define the visible layout and appearance presented to the users, and (2) the non-visual Graph (Widget Blueprint) that manages interaction events. These events act as triggers for invoking specific functions or executing Blueprint logic routines.

The interface was designed in accordance with the principles of simplicity, practicality, and aesthetic. A 512 × 512 chest CT image was positioned on the right side as an information display panel. The left side housed control buttons and text information fields: "Previous" and "Next" buttons were arranged horizontally in the top-left corner, an "Quit" button was positioned in the bottom-left corner, and a text display box was located in the central-left region. Fig. 3 illustrates the overall interface layout and the main events bound to each UI component.

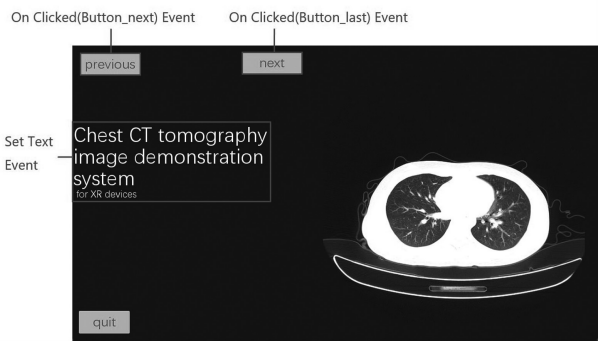

Fig.3 User Interface

The visual construction of MainSurface was performed as follows: First, a widget Blueprint was created using the UMG system, and the UI interface was designed in the Designer module. The

panel was constructed based on a Canvas Panel, integrating interactive components such as buttons, image displays, and text boxes. All components were organized within Overlay widgets to facilitate unified logic implementation and size regulation. To meet text display requirements, Rich Text Blocks were adopted to accommodate potential multi-format text demands. Two Data Tables were established in the project directory to define text styling functions: one specifying the title font format and the other governing the content font format. Subsequently, a Rich Text Block (Rich Text Block0) was added to the UI interface, which inherited the predefined configurations from the Data Tables. Finally, the default text content was edited as appropriate. The UI control logic was implemented within the Blueprint Graph using On Clicked nodes to define button-press functionality. These nodes detect button activation events and execute the corresponding blueprint logic through event-driven mechanisms. As shown in Fig. 4, the Call On Next Button Pressed and Call On Previous Button Pressed nodes were connected to allow the main blueprint to monitor and respond to button-press events. Furthermore, Set Text event nodes were sequentially linked with the target set to the predefined Rich Text Block0 variable (Fig. 5), enabling dynamic modification of text content within the UI rich text box through the main blueprint.

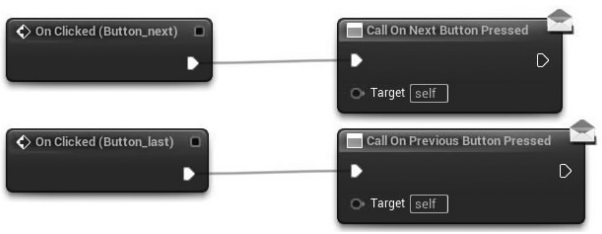

Fig.4 Workflow of the listener button

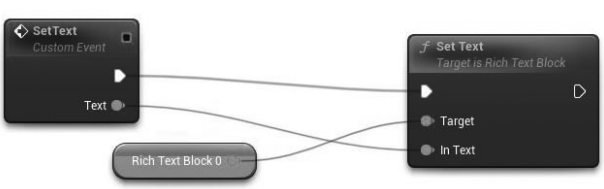

Fig.5 Revised Text Flow

2.2.2 Software Function Coding

To implement the core function of displaying information when hovering the mouse over an image, a mask image must be applied to the original image. The user interface should present the original image, while the underlying logic utilizes the mask image to distinguish different regions. In this way, each region in the mask image correspond to different programmed logic behaviors in the UE, allowing region-specific information to be shown when the user selects different anatomical structures.

2.2.2.1 Initialization

The blueprint initialization logic is illustrated in Fig. 6. The initialization procedure consists of three main components: "Preparing Text Information," "Rendering Images," and "Defining Image Switching Functionality." Among these, the Table Row variable is an array variable that stores row indices from DT_OrganData, whereas Material Instance specifies the surface appearance of the Plane. As system initialization is required, the Event Begin Play node (Begin Event) acts as the trigger to execute the entire initialization logic.

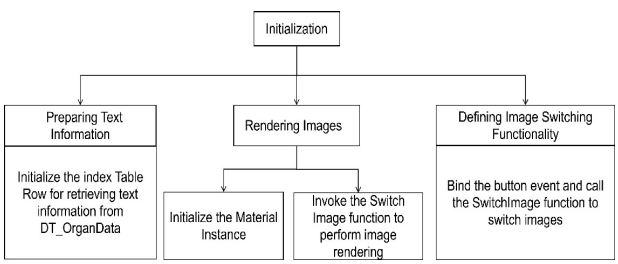

Fig.6 Initialize part of the blueprint logic

(1)Preparing Text Information

First, a Table Row variable is created to store the row indices retrieved from DT_OrganData, and the Get Data Table Row Names function is employed to assign values to this variable.

(2)Image Rendering

A dynamic Material Instance is derived from the parent class material and assigned to the Plane via the Create Dynamic Material Instance node. The Switch Image function is then invoked to render both the original and mask images onto the user interface. When debug mode is disabled, the system displays the original image while hiding the mask image. The parameter Image Index is initialized to 0 by default, indicating that the element at index 0 (the first image) in the Tex-

tures array is rendered.

(3)Defining the Image Switching Functionality

The pointer reference to MainSurface (UI) is obtained via the User Widget base class, enabling the adjustment of parameters such as charts and interface layout within MainSurface without altering the blueprint. The functions "Bind Event to On Next Button Pressed" and "Bind Event to On Previous Button Pressed" functions are utilized to bind the two On Clicked events of the UI, establishing essential references. Only after these event bindings are completed can the blueprint listen for On Click events and respond accordingly to the "Previous" and "Next" button interactions.

Upon clicking the Next Button, the system increments the Current Image Index (initialized to 0) by 1 and passes the updated value as an input parameter to the Switch Image function. If the function returns True (signaling successful rendering), the updated index is assigned back to Current Image Index. This mechanism ensures that each button press not only switches the currently displayed image according to the current index but also advances the index value sequentially, preserving the correct browsing order. All invoked nodes, together with their corresponding inputs and outputs, are listed in Table 2.

Table 2 Initialization of Partial Blueprint Call Nodes with Their Inputs and Outputs

Node	Input	Output	Function
Get Data Table Row Names	DataTable (DT_OrganData)	Table Row	Obtain row names
Create Dynamic Material Instance	Plane, Material	Material Instance	Create material instance
Get Widget	UI	Object	Get control
Cast to MainSurface	Object	MainSurface	Get MainSurface
Addition	Current Image Index	Current Image Index	Auto-increment the Image Index
Switch Image	Current Image Index	Boolean value	Render and switch images
Branch	Boolean value	No output value	Check index availability and debug status
Bind Event to On Next Button Pressed	MainSurface, Next Event	No output value	Bind next event
Bind Event to On Previous Button Pressed	MainSurface, Previous Event	No output value	Bind previous event

2.2.2.2 Main Function

The logic of the main function is shown in Fig. 7, which comprises four components: display speed optimization, acquisition of the RGB values at the cursor position, detection of text refresh requirements via a caching mechanism, and retrieval of textual information from the database.

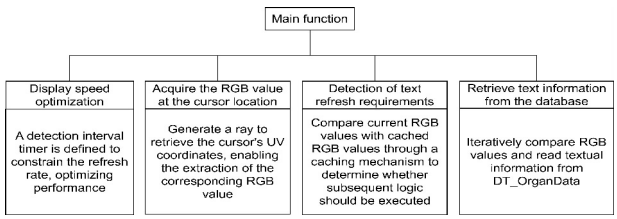

Fig.7 Logic of Main Blueprint Design (Optimized Graph)

(1)Display Speed Optimization

The main logic is triggered by per-frame refresh events. To reduce computational overhead and optimize runtime performance, a timer mechanism is introduced to restrict detection frequency. Experimental results show that setting the interval duration to 0.1 seconds achieves a favorable balance between detection efficacy and computational cost. The timer module utilizes two variables, namely Timer and Interval, where Interval defines the minimum time threshold. At each event update, the Timer increments by Delta Seconds and compares against the Interval value. If Timer exceeds Interval, the system returns True and proceeds with subsequent logic; otherwise, it proceeds to the next frame without further processing.

(2)Acquire the RGB Value at the Cursor Location

The rendering scene utilizes a world coordinate system, which is distinct from both the two-dimensional screen coordinates of the cursor and the UV coordinate system of the render target (RT). Direct acquisition of the cursor's screen coordinates or world coordinates cannot guarantee accurate positions localization on the RT. To extract the RGB values at cursor-specified points on the RT, a mapping relationship between 2D screen coordinates and the world coordinate system is established in this study. Specifically, relevant functions are employed to obtain the world coordinates of both the camera and the cursor. A ray, originating from the camera and directed toward the cur-

sor position, is derived by connecting these two points. Then, the object hit by this ray is filtered to identify the target component. When the ray intersects the Plane, the collision point between the ray and the Plane surface is obtained, from which a corresponding UV coordinate is derived. The Find Collision UV function returns this UV coordinate, which is then decomposed into two 2D vectors (and) using the Break Vector 2D operation.

Since the RT and the Plane share identical dimensions (i. e., matching resolution), this coordinate can be directly mapped to retrieve the corresponding RGB value on the RT. This RGB value is subsequently cross-referenced with the anatomical data stored in DT_OrganData for structural identification. The Read RT UV node processes the UV coordinates together with the RT reference to retrieve the corresponding RGBA values. A conditional Branch evaluates the output of Line Trace By Channel: if false (no intersection detected), it outputs "Trace None" for debugging; if true, the component's name is displayed via Get Display Name to verify RT collision. Concurrently, an additional Branch validates the results of Find Collision UV, outputting "UV Error" upon failure to ensure coordinate acquisition reliability.

(3)Text Refresh Detection (Caching Mechanism)

To optimize system performance and efficiency, a caching variable denoted as Current Color (default: RGBA [0, 0, 0, 0]) is introduced in the subsequent logic. This variable avoids redundant color comparison loops when the cursor remains within a region with uniform RGB values, thus substantially reducing computational overhead. The corresponding workflow is as follows: after obtaining the RGB value at the ray-hit position on the RT, the system first compares it with Current Color via the Color Equal macro. A Branch node then judges this comparison result. If the values match, the subsequent comparison loop is skipped. If they differ, the logic continues while simultaneously setting the Boolean variable Matched to False and updating Current Color to the newly detected color value.

(4)Retrieve Text Information from the Database

The "Row Name" column is extracted from the DT_OrganData table sequentially to form a loop table named Table Row. In each iteration, each "Row Name" is retrieved as a search criterion and compared with the input DT_OrganData in the "Get Data Table Row" function node to obtain the complete row data. Subsequently, the SET node is utilized to decompose this row data into three variables: Organ Name, Color ID, and Text, for use in subsequent logical operations. one of these variables, Color ID (an RGBA value), is compared with the cached color Current Color by invoking the Color Equal macro. A Branch node is employed to determine the outcome of this comparison. If the comparison is successful, a Boolean value of True is output, and the Organ Name is printed and the Text variable is assigned to the Rich Text Block0 in Main Surface. If no matching item is found upon completion of the loop (with Matched remaining False), the text "None" is assigned to the rich-text box in the Main Surface. All invoked nodes, along with their respective inputs and outputs, are enumerated in Table 3.

2.2.2.3 Switch Image Function

The Switch Image function is designed to implement view switching, label identification, the original image display, and the image rendering onto the RT. As shown in Table 4, this function includes one parameter, Image Index (index parameter), which serves as the subscript for the image array (Textures) stored in an array format. Each element within this array contains two images: one original image and one mask image. Based means of this parameter, specific array elements can be accurately located and referenced for subsequent operations.

As illustrated in Fig. 8, the function comprises four modules: Index Checking, Retrieval of Original Image and Mask Images, Texture Rendering to RT, and Material Instance Parameters Setting. All nodes invoked within the function are documented in Table 5.

Table 3 Nodes Called in the Main Blueprint with Their Inputs and Outputs

Node	Input	Output	Function
Addition	Delta Second, Timer	Timer	Timer auto-increment
float > float	Timer, Interval	Boolean value	Timer comparison
Branch	Boolean value	No output value	Timer condition check
Set Timer	Timer	Timer	Set timer variable value
Line Trace By Channel	User Perspective Location, World Derection	Out Hit, Boolean value	Ray construction
Break Hit Result	Out Hit	Object	Classification of hit targets
Get Display Name	Object	String Value	Acquisition of hit target names
Branch	Boolean value	No output value	Hit/Miss determination
Find Collision UV	Out Hit	UV, Boolean value	UV coordinate acquisition
Branch	Boolean value	No output value	Judge whether the UV coordinates are obtained
Breack Vector2D	UV	X, Y	Split the UV coordinates into two separate vectors
Read Render Target Raw UV	RT, X and Y	Liner Color	Acquire the RGBA values
To Color (Linear Color)	Liner Color	FColor	Convert linear colors to full-range colors
Color Equal	Current Color, FColor	Boolean value	Compare the cached color with the current color
Branch	Boolean value	No return value	Determine whether the color has changed
Set Matched	Matched	Mathced	Update the variable Matched
Set Current Color	Current Color	Current Color	Set the cached color
For Each Loop with Break	Table Row	No output value	Loop for applying text information to the Main Surface
Get Data Table Row	DataTable (DT_OrganData)	Row Data	Acquire row information
Color Equal	Current Color, Color ID (Break from Row Data)	Boolean value	Compare the current color with the colors in the data table
Branch	Boolean value	No return value	Determine whether the colors are identical
Set Matched	Matched	Matched	Update the variable Matched
To Text (String)	Text (string)	Text (text)	Convert string to text
Set Text	Main Surface, Text (text)	No output value	Apply text information to Main Surface
Branch	Matched	No output value	Apply the text information 'None' to the Main Surface

Table 4 Parameters of the Switch Image Function

Parameter	Describe
Image Index	Index of the Textures Element in the Texture Array

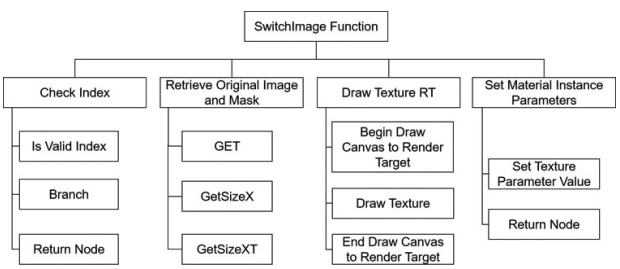

Fig.8 Programming Logic of the SwitchImage Function and Its Key Called Nodes

Table 5 Input and Output Parameters of Internally Called Nodes in the Switch Image Function

Node	Input	Output
Is Valid Index	Image Index, Textrues	Boolean value 1
Branch	Boolean value 1	No output value
Begin Draw Canvas to Render Target	RT	Canvas, Context and Render canvas
GET	Textures, Image Index	Original_Image, Mask
GetSizeX	Target	Return Value (X-coordinate)
GetSizeY	Target	Return Value (Y-coordinate)
Draw Texture	Canvas, Mask	Render texture
End Draw Canvas to Render Target	Context	End texture rendering
Branch	Debug	No output value
Set Texture Parameter Value	Material Instance, Original_Image	Set material instance values
Set Texture Parameter Value	Material Instance, Mask	Set instance material values

(1)Check Index

In this design, array index out-of-bounds errors can easily occur during user interaction. Specifically, when the user repeatedly clicks the "Next" button until the Image Index exceeds the maximum index value of the Textures array (the array variable used for storing images), an out-of-bounds error will be triggered. This error causes the system to throw an exception and interrupt the execution of the logic flow. To address this issue, the Textures array and Image Index are first input into the IS VALID INDEX node to verify the validity of the index, thereby preventing errors associated with array out-of-bounds access. If the index is invalid, the node returns False and terminates the function execution to avoid system malfunctions.

(2)Retrieve Original Image and Mask

Based on the index, the original image and mask image are retrieved from the Textures array. The resolution of the images is then obtained using the GetSizeX and GetSizeY functions.

(3)Draw Texture RT

Following successful index validation, the Begin Draw Canvas to Render Target node is utilized with the RT (texture variable) as input to create a drawing canvas on the RT with the resolution of the input image. Subsequently, the texture is drawn onto the canvas, where the texture source is the mask image retrieved according to the index, and the texture size is determined by the mask image resolution retrieved via the corresponding function. At this stage, although the mask image has been rendered onto the RT, it is not displayed on the screen, since the texture has only been drawn internally without being rendered into the world. This mechanism enables the acquisition of UV coordinates from the mask image while maintaining the display of the original image on the screen.

(4)Set Material Instance Parameters

A Boolean input named Debug is used to control the visibility of the mask image. Two Set Texture Parameter Value nodes are subsequently used: one assigns the original image texture to the Material Instance (a material instance variable), while the other assigns the mask image to the same Material Instance. The Material Instance is initialized in the main blueprint and serves as the only medium through which images can be rendered onto the Plane—a 2D static mesh acting as the rendering panel. The Material Instance is initialized in the main blueprint and The Plane functions as the display carrier for the images, rather than a conventional UI component.

2.2.2.4 Color Equal Macro

In UE, the default color space is linear, with RGBA values represented as floating-point numbers in the range [0,1]. In contrast, externally imported mask images typically adopt the sRGB color space. When implementing RGB recognition logic, directly converting sRGB colors to linear space may lead to precision loss, as the converted linear values cannot guarantee an exact one-to-one mapping with the original sRGB values. This discrepancy can lead to erroneous color identification, where the same color is occasionally recognized as distinct values. Therefore, using converted linear colors directly for color comparison is unreliable. To resolve this issue, the design implements a color space conversion solution and introduces a dedicated macro to handle color comparison operations.

In this process, the Break Color node is used to decompose the two input RGB values into six individual channel components. Each of these six values is then compared for equality on a per-channel basis. A logical AND operation is subsequently applied to combine the three Boolean comparison results, yielding a single Boolean output as the final judgment.

3 Experiment

To quantitatively evaluate the preliminary effectiveness of the proposed system in medical imaging education, an exploratory pilot study was conducted. Ten medical students without prior CT diagnostic experience were recruited and randomly divided into an experimental group ($n=5$) and a control group ($n=5$). The experimental group learned thoracic CT sectional anatomy using the medical image demonstration system developed in this study, while the control group received instruction via pre-recorded 2D CT tomographic tutorial videos (covering both lung window and mediastinal window sequences) combined with 118 static CT images. All participants completed a 20

minutes self-directed learning under standardized environment conditions. Immediately after the learning phase, a standardized identification test was administered. Twenty images were randomly selected from the original CT dataset, each with a specific organ highlighted by red outline. Participants were required to independently identify and record the anatomical name of the outlined structure. The overall identification accuracy for each group was calculated, and a preliminary data analysis of the results was performed using descriptive statistics to assess the potential value and feasibility of the system in educational applications.

4 Results

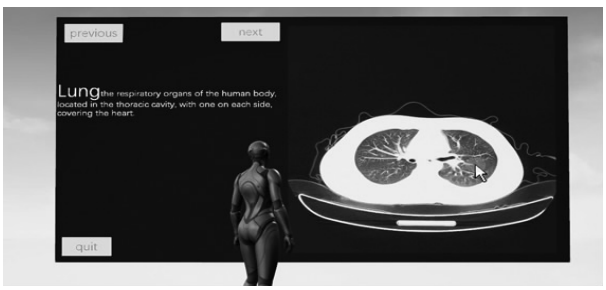

Fig.9 System demonstration results

Based on the methodology described above, this study successfully developed an interactive chest CT image demonstration system compatible with mainstream XR devices, whose rendering results are illustrated in Fig. 9. To evaluate the effectiveness of the systemin medical imaging education, a dedicated comparative experiment was designed. The recognition test results of the experimental and control groups for the lung window and mediastinal window tasks are presented in Table 6 and Table 7, respectively, with the corresponding accuracy rates summarized in Table 8 and Table 9. Quantitative analysis indicates that the experimental group achieved an accuracy of 82% for the lung window task, 88% for the mediastinal window task, and an overall accuracy of 85%. In contrast, the control group exhibited accuracies of 50%, 44%, and 47% for the same tasks. These results demonstrate the significant potential advantage of the proposed system in improving the recognition accuracy of thoracic anatomical structures.

Table 6 Lung Window Test Results

Subjects	Participating Group	Right Lung	Trachea	Left Bronchus 1	Left Bronchus 2	Heart 1	Heart 2	Base of Lung	Colon 1	Colon 2	Small Intestine
1	Experimental Group	√	√	√	√	√	√	√	√	√	√
2	Experimental Group	√	√	√	√	√	√	√	√	√	√
3	Experimental Group	√	√	√	√	√	√	√	√	√	√
4	Experimental Group	√	√	×	×	×	×	×	√	√	×
5	Experimental Group	√	√	√	×	√	√	√	√	×	×
6	Control Group	√	√	×	×	×	×	√	×	×	×
7	Control Group	√	√	×	√	√	√	√	×	×	×
8	Control Group	√	√	√	×	×	×	√	×	×	×
9	Control Group	√	√	√	×	×	×	√	×	×	×
10	Control Group	√	√	√	×	√	√	√	√	√	×

Table 7 Mediastinal Window Test Results

Subjects	Participating Group	Spinal Cord	Left Lung	Left Bronchus	Heart	Liver 1	Liver 2	Colon	Spleen	Kidney	Small Intestine
1	Experimental Group	√	√	√	√	√	√	√	√	√	√
2	Experimental Group	√	√	√	√	√	√	√	√	√	√
3	Experimental Group	√	√	√	√	√	√	√	√	√	×
4	Experimental Group	√	√	×	√	√	√	√	×	√	√
5	Experimental Group	√	√	×	√	√	×	√	√	√	×
6	Control Group	√	√	×	×	×	×	×	×	√	√
7	Control Group	√	√	×	√	√	√	×	√	√	×
8	Control Group	√	√	√	×	×	×	×	×	×	×
9	Control Group	√	√	√	×	×	×	×	×	×	×
10	Control Group	√	√	×	√	×	×	√	×	√	×

Table 8 Experimental Group Recognition Accuracy Rate(%)

Participating Group	Lung Window Recognition Accuracy Rate	Mediastinal Window Recognition Accuracy Rate	Overall Accuracy Rate
1	100	100	100
2	100	100	100
3	100	90	95
4	40	80	60
5	70	70	70
Mean	82	88	85

Table 9 Control Group Recognition Accuracy Rate(%)

Participating Group	Lung Window Recognition Accuracy Rate	Mediastinal Window Recognition Accuracy Rate	Overall Accuracy Rate
6	30	40	35
7	60	70	65
8	40	30	35
9	40	30	35
10	80	50	65
Mean	50	44	47

5 Discussion

This study presents an interactive chest CT imaging demonstration system based on extended reality (XR) technology. By significantly improving interactivity, spatial perception accuracy, and cross-platform compatibility, the proposed system effectively addresses the limitations of conventional medical imaging teaching platforms and preoperative planning communication systems. Experimental results demonstrate that the experimental group using this system achieved higher recognition accuracy than the control group using traditional teaching methods, which is consistent with previous studies verifying the effectiveness of XR in enhancing medical education[24−26].

Compared with existing technologies, the innovation and advantages of the proposed system are mainly reflected in the following aspects. In contrast to traditional PACS teaching systems[27−28], which rely on browsing 2D tomographic image sequences and involve high cognitive abstraction, our system provides learners with an intuitive and interactive human structural model through a 3D spatial interaction interface and real-time dynamic feedback mechanisms. This design effectively improves the spatial comprehension and memorization efficiency of complex anatomical relationships. In terms of system architecture and data compatibility, our system overcomes the limitation of commercial VR anatomy software such as 3D Organon VR Anatomy[29], which only offers predefined standardized models and lacks support for importing and processing real patient DICOM data. The standardized multi-modal image integration framework established in this study enables seamless conversion from raw DICOM data to immersive scenes, supporting the rapid deployment of multi-modal imaging modalities including MRI and PET-CT by simply inputting data and corresponding annotation masks. This provides crucial support for teaching and preoperative planning based on real clinical cases. Regarding technical implementation and dissemination potential, compared with dedicated teaching applications developed using Unity (e. g., Diegritz *et al*.[30]), our system takes advantage of Unreal Engine's high-performance rendering and cross-platform architecture. It not only ensures cinematic-grade visual quality and smooth operation but also exhibits superior hardware compatibility and deployment flexibility. The system can adapt to a wide range of hardware configurations, from high-end PC VR setups to mobile all-in-one devices. This directly alleviates the cost and accessibility barriers of deploying immersive technologies in resource-limited environments, thereby improving the overall technical accessibility.

Despite the aforementioned advancements, this study has certain limitations: (1) The preparation of mask images relies on manual annotation, and the integration of automated segmentation algorithms requires further exploration; (2) The current system is primarily optimized for static CT/MRI images, with inadequate support for dynamic imaging modalities (e. g., echocardiography). Future research will focus on developing fully automated segmentation models to minimize manual intervention and expanding the system's capabilities to support real-time rendering and interactive analysis of 4D dynamic medical imaging. Building on this foundation, a large-scale, multi-center ran-

domized controlled trial is planned to further validate the effectiveness and clinical applicability of the proposed system in medical education and preoperative planning scenarios.

6 Conclusions

This study successfully developed an XR-based interactive teaching system for thoracic CT imaging. Through innovative three-dimensional interaction design and real-time information feedback mechanisms, the system effectively overcomes the limitations of traditional medical imaging teaching platforms in terms of spatial perception and interactivity. Experimental results validate that the system improves learners' recognition accuracy of thoracic anatomical structures, thus demonstrating its effectiveness and practical value.

Declaration of competing interest

We declare that we have no conflict of interest.

Funding

Funding: This research was supported by the Nanchong Science and Technology Planning Project Fund (Grant No. 22YYJCYJ0068) and the Sichuan Provincial Department of Science and Technology (Project No.: 2023NSFSC0646).

Ethics approval and consent to participate

The study has been approved by the Ethics Committee of North Sichuan Medical College. Due to the retrospective nature of the research, informed consent is not required.

References

[1] Chen GH. Clinical applications of X-ray computed tomography (CT): A review[J]. Chinese Journal of Medical Instrumentation, 1978(3): 58－67.

[2] Wan ML. Comparative analysis of the diagnostic value of CT and X-ray imaging for femoral head avascular necrosis[J]. World Latest Medicine Information, 2016, 18(16): 181－182.

[3] Huang X, Gu H, Wu R, *et al*. Chest imaging classification in Mycoplasma pneumoniae pneumonia is associated with its clinical features and outcomes[J]. Respiratory Medicine, 2024, 221: 107480.

[4] Li Z, Zeng B, Lei P, *et al*. Differentiating pneumonia with and without COVID-19 using chest CT images: from qualitative to quantitative[J]. Journal of X-Ray Science and Technology, 2020, 28(4): 583－589.

[5] Jiang F, Xu C, Wang Y, *et al*. A CT-based radiomics analyses for differentiating drug-resistant and drug-sensitive pulmonary tuberculosis[J]. BMC Medical Imaging, 2024, 24(1): 307.

[6] Nijiati M, Guo L, Abulizi A, *et al*. Deep learning and radiomics of longitudinal CT scans for early prediction of tuberculosis treatment outcomes[J]. European Journal of Radiology, 2023, 169: 111180.

[7] Ma JK, Zhou S, Wang WQ, *et al*. Evaluating the diagnostic value of "Emphysema Index" and "Emphysema Distribution Patterns" in quantitative HRCT of the chest for asthma-COPD overlap (ACO)[J]. Journal of Imaging Research and Medical Applications, 2021, 5(10): 2.

[8] Yuan K, Qin W, Yang GQ, *et al*. The guiding role of preoperative enhanced chest CT in bronchial artery embolization for bronchiectasis patients with massive hemoptysis[J]. Shandong Medical Journal, 2024, 64(23): 76－78.

[9] Liang LD, Wan L, Liu Y, *et al*. Differential diagnostic value of chest CT enhanced scan and MRI examination in benign and malignant anterior mediastinal tumor[J]. Oncology Progress, 2024, 22(15): 1693－1696

[10] Sun ZH. A case report of multiple aortic aneurysms[J]. Journal of Practical Medical Techniques, 2011, 18(6): 645.

[11] Qu N, Wang XY, Luo YH, *et al*. The CT diagnosis of diffuse plueral mesothelioma[J]. Journal of Clinical Radiology, 2010, 29(6): 832－834.

[12] Liao K, Ye B, Li X, *et al*. Clinical application of third-generation dual-source CT-based dynamic imaging reconstruction for pulmonary embolism imaging[J]. Journal of Cardiothoracic Surgery, 2025; 20(1): 86.

[13] Zhang Y. Diagnostic value of chest CT plain scan combined with D-dimer for suspected aortic dissection in emergency patients with chest pain[J]. Guide of China Medicine, 2019, 17(10): 131.

[14] Zeng S, Chen L, Lan S. Research on the extension of respiratory interaction modalities in virtual reality technology and innovative methods for healing anxiety disorders[J]. Scientific Reports, 2025, 15: 7936.

[15] Zhang YT. Development of first person shooting games based on UE5[J]. Modern Computer, 2023, 29(18): 113－116.

[16] Zhang YH, Ding YD, Cai JY. The matrix awakens: A new aesthetic experience in UE5-based image production[J]. Advanced Motion Picture Technology, 2022(11): 54－58.

[17] Li LQ, Chen QQ, Zhou WW, *et al*. Analysis of the application of post-processing techniques in CT nedical imaging teaching [J]. Journal of Imaging Research and Medical Applications, 2021, 5(20): 227－228.

[18] Fugate JMB, Tonsager MJ, Macrine SL. Immersive extended reality (I-XR) in medical and nursing for skill competency and knowledge acquisition: a systematic review and implications for pedagogical practices[J]. Behavioral Sciences (Basel, Switzerland), 2025, 15(4): 468.

[19] Oyama S, Iwase H, Yoneda H, *et al*. Insights and trends review: Use of extended reality (xR) in hand surgery[J]. The Journal of Hand Surgery, European Volume, 2025, 50(6): 762－770.

[20] Hess CW, Rosenbloom BN, Mesaroli G, *et al*. Extended reality

(XR) in pediatric acute and chronic pain: systematic review and evidence gap map[J]. JMIR Pediatrics and Parenting, 2025, 8: e63854.

[21] Chen FQ, Leng YF, Ge JF, *et al*. Effectiveness of virtual reality in nursing education: meta-analysis[J]. Journal of Medical Internet Research, 2020, 22(9): e18290.

[22] Efendi D, Apriliyasari RW, Prihartami Massie JGE, *et al*. The effect of virtual reality on cognitive, affective, and psychomotor outcomes in nursing staffs: systematic review and meta-analysis[J]. BMC Nursing, 2023, 22(1): 170.

[23] Mondal H, Mondal S. Adopting augmented reality and virtual reality in medical education in resource-limited settings: constraints and the way forward[J]. Advances in Physiology Education, 2025, 49(2): 503－507.

[24] Huai P, Li Y, Wang X, *et al*. The effectiveness of virtual reality technology in student nurse education: A systematic review and meta-analysis[J]. Nurse Education Today, 2024, 138: 106189.

[25] Neher A N, Bühlmann F, Müller M, *et al*. Virtual reality for assessment in undergraduate nursing and medical education - a systematic review[J]. BMC medical education, 2025, 25(1): 292.

[26] He Y, Wang Z, Sun N, *et al*. Enhancing medical education for undergraduates: integrating virtual reality and case-based learning for shoulder joint[J]. BMC medical education, 2024, 24(1): 1103.

[27] Rita M, Aeen M, Rajabzadeh F, *et al*. Using PACS for teaching radiology to undergraduate medical students[J]. BMC Medical Education, 2024, 24(1): 935.

[28] Bitar I, Ghannam J, Nandalur K, *et al*. Comparative analysis of scrollable DICOM images and static CT images in teaching thoracic imaging anatomy to first-year medical students[J]. Anatomical Sciences Education, 2025, 18(6): 604－608.

[29] Weyant EC, Woodward NJ. 3D organon VR anatomy: A virtual anatomy medical education tool[J]. Journal of Electronic Resources in Medical Libraries, 2021, 18(4): 198－203.

[30] Diegritz C, Fotiadou C, Fleischer F, *et al*. Tooth Anatomy Inspector: A comprehensive assessment of an extended reality (XR) application designed for teaching and learning of root canal anatomy by students[J]. International Endodontic Journal, 2024, 57(11): 1682－1688.

(Received: November 12. 2025 **Revised**: December 28. 2025)