Received: 29 January 2021 | Revised: 22 April 2021 | Accepted: 27 April 2021
DOI: 10.1002/ca.23755

MEDICAL AND DENTAL EDUCATION

CLINICAL ANATOMY WILEY

Embedding interactive, three-dimensional content in portable document format to deliver gross anatomy information and knowledge

Jon Jatsu Azkue

Department of Neurosciences, School of Medicine and Nursery, University of the Basque Country, UPV/EHU, Leioa, Bizkaia, Spain

Correspondence
Jon Jatsu Azkue, Department of Neurosciences, School of Medicine and Nursery, University of the Basque Country, UPV/EHU, Sarriena Auzoa z/g, 48940 - Leioa, Bizkaia, Spain.
Email: jonjatsu.azkue@ehu.eus

Abstract

The Portable Document Format (PDF) is likely the most widely used digital file format for scholarly and scientific electronic publishing. Since format specification version 1.6, three-dimensional (3D) models in Universal 3D (U3D) format can be embedded into PDF files. The present study demonstrates a repertoire of graphic strategies and modes of presentation that exploit the potentials of 3D models embedded in PDF to deliver anatomical information and knowledge. Three-dimensional models and scenes representing anatomical structures generated by 3D surface scanning or by segmentation from either clinical imaging data or cadaver sectional images were converted into U3D format and then embedded into PDF files using both freely and commercially available software. The relevant steps and required software tools are described. Built-in tools in *Adobe Acrobat* and *JavaScript* scripting both were used to pre-configure user interaction with 3D contents. Eight successive proof-of-concept examples of increasing complexity are presented and provided as supplementary material, including both unannotated and annotated 3D specimens, use of bitmap-textures, guided navigation through predetermined 3D scenes, 3D animation, and interactive navigation through tri-planar sectional human cadaver images. Three-dimensional contents embedded in PDF files are generally comparable to multimedia and dedicated 3D software in terms of quality, flexibility, and convenience, and offer new unprecedented opportunities to deliver anatomical information and knowledge.

KEYWORDS
3D visualization, medical education, multimedia, surface rendering

1 | INTRODUCTION

Acquisition of anatomical knowledge is essentially a three-dimensional (3D) endeavor, largely concerned with both identifying bodily structures and understanding their spatial organization. Traditionally, cadaver dissection has been integral to anatomy education, as it provides unparalleled opportunities to appreciate 3D relationships as well as the range of variability present in the real human body (Aziz et al., 2002; McLachlan, Bligh, Bradley, & Searle, 2004; Older, 2004). Among the various complementary methods that have been incorporated over the last decades, 3D computer visualization of digital anatomical models has proven to be effective in anatomy learning in terms of both factual and spatial knowledge (Murgitroyd, Madurska, Gonzalez, & Watson, 2015; Triepels et al., 2020; Yammine &

Violato, 2015). Three-dimensional visualization can be seen as a useful adjunct where there are limitations to laboratory training time and access to cadaveric material due to financial, ethical, religious, or safety considerations (McLachlan et al., 2004; Notzer, Zisenwine, Oz, & Rak, 2006; Raja & Sultana, 2012; Sugand, Abrahams, & Khurana, 2010). Most recently, moreover, 3D virtual anatomy platforms have become a major learning resource in the switch to distance learning methods adopted by higher education institutions on account of the Covid-19 pandemic (Longhurst et al., 2020).

At the present time, the *Portable Document Format* (PDF) is probably the most widely used digital file format for electronic publishing and exchange of scholarly and scientific documents (Chapman, Turland, & Watson, 2010; Hitchcock, Carr, & Hall, 1997). The PDF is a digital document description format developed by Adobe Systems Inc. (Adobe, San Jose, CA) to define the contents and formatting of textual and graphic elements in electronic documents, irrespective of the software, hardware, and operating systems. Portable Document Format files are typically viewed with *Adobe Acrobat Reader* software (hereinafter *Acrobat Reader*), although other viewer applications also have been made available after PDF was standardized as ISO 32000 (Adobe Systems Inc., 2008). Nonetheless, most non-Acrobat viewer applications cannot render 3D scenes.

In 2005, *Acrobat Reader 7* and *Acrobat 7 Professional* included the *3D annotations* functionality to embed 3D models in U3D (Universal 3D) file format into PDF files as an implementation of the PDF format specification version 1.6 (Adobe Systems Inc, 2004). The 3D contents so presented may include point clouds, lines, or polygon meshes defined in a 3D space that the user can explore interactively by zooming, panning, and rotating the scene, as well as by manipulating visibility of selected objects. The U3D format also supports animation, and can be annotated using the *Acrobat* built-in *3D comment* functionality. Moreover, properties of embedded 3D models such as appearance and spatial coordinates can be accessed and manipulated using *JavaScript* cross-platform scripting language, thus providing substantial control over the embedded 3D content (Adobe Systems Inc., 2015). Due to its flexibility and versatility, authors have advocated for adopting 3D-PDF technology as a de facto standard for electronically publishing biomedical study outcomes (Barnes et al., 2013; Danz & Katsaros, 2011; Murienne, Ziegler, & Ruthensteiner, 2008; Newe & Becker, 2018; Ruthensteiner & Hess, 2008).

Although the use of interactive 3D PDF files is not yet widely adopted, a recent review identified embedded 3D content in over 200 articles from a variety of disciplines, ranging from astronomy to animal and human surgery, either as a figure included within the article body or as an external reference (Newe & Becker, 2018). In those with a focus on human anatomy, presentation of interactive 3D content was largely based on surface representations of bodily structures, either alone (Kato et al., 2014; Kato, Ziegler, Utsumi, Ohno, & Takeichi, 2016; Chung, Chung, & Park, 2015; Chung, Kwon, Shin, & Chung, 2017; Chung, Chung, Shin, & Kwon, 2018; Kim et al., 2013, Kim, Chung, Park, Shin, & Park, 2015; Park, Chung, Shin, Jung, & Park, 2015; Shin et al., 2015; Wu et al., 2017) or in combination with sectional anatomy (Chung, Chung, & Park, 2015; Jang, Chung, & Shin, 2015; Reina, Lirk, Puigdellívol-Sánchez, Mavar, & Prats-Galino, 2016; Shin et al., 2012, 2012; Shin, Kwon, et al., 2015; Shin, Lee, Park, Lee, & Chung, 2015; Shin, Shim, & Kim, 2018a, 2018b; Valera-Melé et al., 2018), and less commonly accompanied by additional or custom-designed graphic user controls (Reina et al., 2016; Valera-Melé et al., 2018).

Despite the many opportunities that 3D-PDF technology offers for presenting anatomical information and knowledge interactively, such potential has been little explored in the context of anatomical knowledge delivery and particularly in the scope of teaching and learning human anatomy.

The aim of the present report was to demonstrate a repertoire of graphic strategies and modes of presentation that exploit the potentials of 3D graphics embedded in PDF files to deliver anatomical information and knowledge, using both built-in tools in *Adobe Acrobat* and *JavaScript* scripting for pre-configuring and customizing 3D content presentation and user interaction. Proof-of-concept examples of interactive 3D-PDF files are provided as supplementary material.

2 | MATERIALS AND METHODS

2.1 | Workflow and software

Producing an interactive 3D-PDF file involves (a) the generation of a U3D file describing the 3D content, (b) embedding such U3D file in a PDF file, and (c) configuring viewing and user interaction options, if required. Although the latter two may be accomplished using *Acrobat*, other solutions are also available. Software tools used in the present work are summarized in Table 1. The basic workflow is shown in Figure 1, and a brief account of the production process is provided as follows.

2.1.1 | Generation of a 3D mesh and production of a U3D file

The process of 3D-PDF file production begins with creating a 3D polygon mesh or a set of meshes that are to be displayed in a 3D scene. Three-dimensional meshes representing anatomical structures can be easily generated by 3D-scanning of an actual specimen, with or without a bitmap-texture (e.g., *Example 1*), or via segmentation of a structure of interest from a sectional image stack of clinical imaging scans (*Examples 2 through 4*, and *8*) or cadaver cross-sections (*Examples 5* and 7). Three-dimensional meshes are usually exported to standard file formats such as STL, OBJ, PLY, X3D or VTK, among others, that describe spatial geometry and provide support, to varying extents, to a range of additional functionalities such as lighting, appearance or material, textures, or animation. Irrespective of the production route, a 3D-mesh or a set of meshes all need to be converted into U3D format before they can be embedded in a PDF file. Conversion may be accomplished using a variety of software applications

TABLE 1 Software used in the process of production of interactive 3D-PDF files

Sub-process	Denomination	Availability	Platform	Website
2D image pre-processing	GNU Image Manipulation Program	Free, open source	MS Windows, MacOS, GNU Linux	https://www.gimp.org
Segmentation	ITK-SNAP Medical Image Segmentation Tool	Free, open source	MS Windows, MacOS, GNU Linux	https://www.itksnap.org
U3D generation	Deep Exploration v6.3 (CAD Edition, Right Hemisphere)	Commercial	MS Windows	
	MeshLab	Free, open source	MS Windows, MacOS, GNU Linux	https://www.meshlab.net
	IDTFConverter	Free, open source	MS Windows	https://sourceforge.net/projects/u3d/
PDF embedding	Adobe Acrobat Pro 2017	Commercial	MS Windows, MacOS	
	LaTeX, *Movie 15* package	Free, open source	MS Windows, MacOS, GNU Linux	https://ctan.org/pkg/movie15
	LaTeX, *Media 9* package	Free, open source	MS Windows, MacOS, GNU Linux	https://ctan.org/pkg/media9
3D postprocessing	Blender	Free, open source	MS Windows, MacOS, GNU Linux	https://www.blender.org

Note: Denominations of the software packages utilized for each sub-process in the production pipeline are indicated, including alternative, freely available software tools. Operating system platforms are indicated for which binary installation files are currently available for download, as well as (commercial or free) availability and corresponding official website addresses.

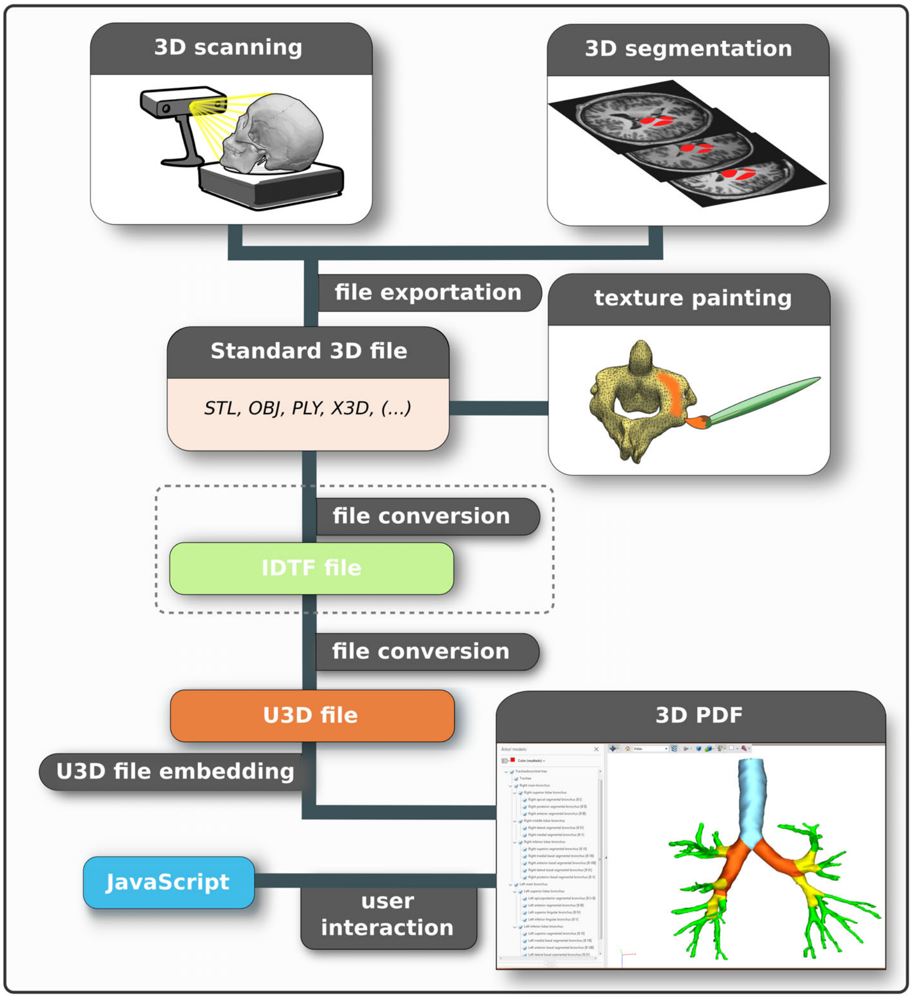

FIGURE 1 Flow diagram for the production of interactive PDF files with embedded 3D content. Three-dimensional contents generated by surface scanning or by segmentation from stacks of clinical imaging data or cadaver cross-sectional images are first exported to standard 3D file formats including, but not restricted to, STL, OBJ, PLY, or X3D. Texture painting can then be applied, if required, to polygon mesh surfaces stored in a file format supporting bitmap-texture data, for example, the OBJ format. Three-dimensional content comprising single or multiple models is then packed into a single U3D file format, either directly or through an intervening IDTF file, and the U3D file is ultimately embedded in a PDF file. In addition to basic user interaction and visualization options of the 3D content, which can both be pre-configured by the document creator, additional custom-designed functionalities can be added via *JavaScript* scripting

freely or commercially available. Here, both proprietary *Deep Exploration v6.3* software (CAD Edition, Right Hemisphere$^{®}$) and open source *MeshLab v2020.07* software (Cignoni et al., 2008) were used to generate U3D files (indicated where appropriate). It is noteworthy that a human-readable *Intermediate Data Text File* (IDTF) can be generated in the U3D file conversion process (e.g., using the *MeshLab* software) as an intermediate optional step prior to transformation into the binary U3D format. This text-based format allows the document creator to edit a variety of features of the 3D scene such as shaders and lighting, the names and appearances of meshes, as well as their organizational hierarchy or structure tree, according to the IDTF format description by *Intel* (Intel Corporation, 2005). Conversion from IDTF into U3D can be accomplished utilizing the *IDTFConverter* utility (Table 1).

2.1.2 | Embedding U3D content in a PDF file

A three-dimensional scene contained in a U3D file can be inserted into a PDF file in various ways. The *Adobe Acrobat Pro 2017* package provides a venue for creating a new PDF file directly from a U3D file, as well as for inserting U3D as *Rich Media* in a preexisting document. In addition, CAD or 3D modeling software applications such as *SolidWorks* or *Rhino 3D* include 3D-PDF files as an export option. Although perhaps in a somewhat less convenient way, 3D-PDF files also can be produced using the freely available *LaTex* typesetting system (https://www.latex-project.org). A *LaTex* editor usually generates a PDF file by compiling source code that contains both text and formatting instructions. Among the over 4000 so-called *packages* currently provided by the *LaTex* support community, that is, modules that can be downloaded and used immediately to add extra functionalities, at least two support multimedia contents, including 3D objects, namely packages *Movie 15* and *Media 9*. By using any of these packages, source code referencing a U3D file can be compiled to create a 3D-PDF file using *LaTeX*. Interestingly, *LaTeX* code is generated automatically as a TEX file by *MeshLab* upon exporting a U3D file, which invokes the *Movie 15* package to embed U3D contents into a PDF file. Although the *Movie 15* package has been deprecated and superseded by *Media 9*, the latest version of *MeshLab* (v2020.12d, built December 2020) still makes use of the former. Sample *LaTex* code for 3D-PDF file generation is provided in Appendix I along with code adapted for the current *Media 9* package, that can be used as a template.

2.1.3 | Configuring view and user interaction in acrobat

Acrobat Reader provides tools that allow the user to control a range of viewing parameters in the 3D scene, including switching among predetermined views, lighting, or model rendering modes, showing/hiding mesh nodes, or changing background color. Rendering and visualization options in *Acrobat Reader* have been described elsewhere (Barnes et al., 2013; Tabernero Rico, Juanes Méndez, & Prats-Galino, 2019) and will not be addressed here. In addition, the *Acrobat Professional* package provides the document creator with powerful tools to preconfigure both the 3D scene and user interaction. The *Manage views…* option in the *Views* drop-down menu allows to set default and secondary views the user will find available to switch among using *Acrobat Reader*. This feature can be made use of to arrange an orderly content presentation (see *Example 6* below). *Acrobat Professional* also includes *3D-comments* as a mechanism to anchor annotations to specific spatial coordinates on model surfaces. In addition, a *JavaScript* application programming interface specific to 3D contents in *Acrobat* (*JavaScript TM for Acrobat 3D Annotations API*; Adobe Systems Inc., 2015) is available for the document creator to dynamically access and manipulate embedded U3D contents (*Examples 7* and 8). *JavaScript* is a cross-platform scripting language through whose extensions *Acrobat* provides access to much of its functionality, including custom-defined effects on embedded 3D contents.

2.2 | Generation of example 3D-PDF files

In the following section, a brief account is given of the specific methods used for generating the interactive 3D-PDF files presented in the *Results* section.

2.2.1 | Example 1

A dry specimen of a human axis vertebra from the anatomy lab of the School of Medicine and Nursery at the University of the Basque Country—UPV/EHU—was surface scanned using a desktop 3D-scanner (Einscan SE, Shining 3D Technology GmbH, Stuttgart, Germany) to produce a textured surface mesh. In the present context, a *texture* is a 2D bitmap that is wrapped onto the mesh surface using specific spatial coordinates in order to attach its color information to 3D surface geometry. The scanned specimen was exported in OBJ format, a process which generated two additional files, namely (a) an MTL file specifying the synthetic material assigned to the mesh and providing a reference to the texture bitmap, and (b) the bitmap file itself in standard JPG format. *MeshLab* software was utilized to generate a U3D file, which was then imported to *Adobe Acrobat Pro 2017* to produce the final 3D-PDF file.

2.2.2 | Example 2

A Beauchêne skull preparation was disassembled and CT scanned at 1 mm thickness, and the resulting DICOM files were imported to *ITK-SNAP Medical Image Segmentation Tool* software (v3.4.0; Yushkevich et al., 2006) for semi-automatic 3D segmentation using the active contour algorithm (Kass, Witkin, & Terzopoulos, 1988). Surface geometries of individually segmented skull bones were exported as separate polygon meshes in STL (*Standard Tessellation Language*) format, and *Blender v2.76* software was utilized for mesh postprocessing, which included polygon number reduction and surface smoothing. *Blender* was also used to partially reassemble a virtual skull based on the digitized polygon meshes. The assembly, which comprised the digital replicas of the occipital, ethmoid, sphenoid, and vomer bones, the left half of the frontal bone cut in the median plane, and the left one of each paired bone, was packaged into a U3D file using *Deep*

Exploration and then imported to *Adobe Acrobat Pro 2017* to produce the final 3D-PDF file.

2.2.3 | Example 3

A dry specimen of a human left temporal bone was 3D-scanned to produce a nontextured 3D surface mesh, that is, geometry data with no associated color information. Three-dimensional arrows were created with *Blender* and arranged spatially as close to and pointing towards anatomical features on the surface of the digitized bone. Pipe-like surface meshes also were created and placed traversing bony canals and in apposition to vascular grooves. *Deep Exploration* software was used to assign synthetic colors to both the temporal bone mesh and all generated 3D glyphs, the latter of which were grouped as *Markers*, and to finally generate a single final U3D file. After importing the resulting U3D file to PDF, captions were inserted pointing to appropriate anatomical structures in four standard views, using the built-in *3D-comment* functionality of *Adobe Acrobat Pro 2017*. Thus, two different mechanisms were resorted to here to identify or annotate morphological features of interest on the digitized bone, namely (a) 3D glyphs (arrows and pipe-like elements) incorporated to the scene *prior to* PDF embedding, and (b) captions inserted within the PDF using the *3D-comment* tool. The criterion to annotate using either 3D glyphs or 3D-comments was that of convenience for the mere purpose of method illustration.

2.2.4 | Example 4

A dry specimen of a human mandible was surface scanned to produce a nontextured 3D surface mesh, and the resulting file was imported in *Blender* software for texture painting. Texture painting is the process whereby colors are applied to a bitmap texture wrapping a 3D surface mesh, in such a manner that the surface will be rendered as having been painted on. Texture painting was used here to delineate the sites of muscle insertion on the mandible. The resulting textured 3D mesh was exported in OBJ format, converted into U3D using *MeshLab* and finally embedded in a 3D-PDF file using *Adobe Acrobat Pro 2017*.

2.2.5 | Example 5

Three-dimensional surface meshes representing the lungs, pulmonary arteries, and the tracheobronchial tree were reconstructed from axial plane sections of the thorax (sections 1290 through 1540, at 1 mm intervals) of the male cadaver from the *Visible Human Project* dataset (National Library of Medicine, Bethesda, Maryland; Spitzer, Ackerman, Scherzinger, & Whitlock, 1996; Ackerman, 1998) by manual segmentation using *ITK-SNAP*. *Blender* was then used to smooth reconstructed meshes, reduce polygon number, and dissect out the trachea and segmental, lobar, and main bronchi from the whole tracheobronchial tree mesh. Surface meshes of each category were grouped in a hierarchical structure using *Deep Exploration*, based on the relationships of each bronchus to parent and children bronchi. Anatomical terms and structural hierarchy both were based on the taxonomy defined by the *Teminologia Anatomica* (FCAT, 1998). Colors were assigned in keeping with the above hierarchy, and the lung model was assigned a low opacity level (30%) in order to be rendered as faintly delineating the lung contours for spatial reference. All meshes were then packed into a single U3D file and finally imported to *Adobe Acrobat Pro 2017* to generate the final 3D-PDF file.

2.2.6 | Example 6

For this example, 3D models from the *Visible Ear* application developed at the Eaton-Peabody Laboratory of Auditory Physiology (Massachusetts Eye & Ear Infirmary, Harvard Medical School) were used, with permission. Briefly, 3D models of the middle, inner and outer ear in their surgically relevant surroundings were produced for the *Visible Ear Model* (https://www.masseyeandear.org/research/otolaryngology/eaton-peabody-laboratories/visible-ear-model) by manual segmentation of serial 25 μm-thick sections of a frozen right human ear of a 14-year-old male adolescent (Wang, Merchant, & Sorensen, 2007). Here, all 3D models were colored individually and packed to a single U3D file using *Deep Exploration*. After importing U3D content to *Adobe Acrobat Pro 2017*, five sequential views named *Scene #1* through *#5* were created with the aid of the *Manage views…* option on the *Views* drop-down menu. Each scene was intended to represent a different stage along a virtual dissection-like process starting from viewing the specimen as a whole and progressively locating and exposing the constituent structures of the middle and inner ear. In this process, each new *View* can be configured to retain camera properties, background color, render mode, lighting, and mesh visibility. Removing anatomical structures from the successive scenes was accomplished here simply by manipulating the visibility of the corresponding mesh node using the *Model tree* control. Please note that this control may be named differently in localized versions of Acrobat Reader and in 3rd party PDF readers other than Adobe's.

2.2.7 | Example 7

Three hundred and forty six axial plane sections of the head (sections 1005 through 1350) from the frozen male cadaver of the *Visible Human Project* dataset were imported to *ITK-SNAP* software in order to create a volumetric pixel dataset. A volumetric pixel or *voxel*, that is, a data element that represents a specific grid value in a 3D space, is created by assigning each pixel of the image stack a dimension in the Z-axis. A set of volumetric data can then be resliced in the X- and Y-axes to generate sagittal and coronal sections, respectively. Thus, 26 equally spaced axial plane sections, 25 coronal plane sections, and 19 sagittal plane sections were extracted from the volumetric dataset and exported in PNG image format. Extracted images were cropped to the head dimensions in each space axis, and downsampled using the *GNU Image Manipulation Program* (GIMP). Axial, coronal, and sagittal section images were then mapped as textures to a set of 68 plane meshes generated in the corresponding space coordinates using *Blender* software. All these bitmap-textured planes were exported individually as OBJ files and all packed to a single U3D file using *Deep Exploration*. After importing the resulting U3D file to *Acrobat Pro 2017*, a simple user interface was created on the left margin of the viewport by adding square-shaped 2D fields that were then enabled via *JavaScript* code to react as buttons. These included buttons to (a) navigate from one section to the next in ascending or descending order, (b) select the axis of navigation, and (c) return to the default tri-planar view.

2.2.8 | Example 8

Surface models of the skull and mandible were generated by semi-automatic segmentation from axial CT scans of the male cadaver from the *Visible Human Project* dataset using *ITK-SNAP*. The two surface meshes were exported separately in STL format and converted into U3D files using *MeshLab*. After importing the U3D files to *Acrobat Pro 2017*, *JavaScript* was utilized to configure and execute mobilization of the surface mesh representing the mandible within the 3D scene by manipulating its spatial coordinates, so as to recreate hinge and translation movements of the condyle relative to the mandibular fossa. *JavaScript* was used to define (a) the desired extent of mandibular rotation and translation, (b) the number of fractional movements—steps or animation *frames*—to complete the animation, and (c) the time allocated to each step (cf. Appendix II for a sample script). The latter two parameters allow the document author to configure smoothness and speed of the animation. In order to enable the user to trigger animation, a button was created that linked mouse input with *JavaScript* code responsible for mobilizing the mandible mesh.

3 | RESULTS

Eight 3D-PDF files with varying degrees of complexity were created that illustrate distinct strategies for delivering interactive 3D content for anatomical visualization. The presented repertoire includes both unannotated and annotated 3D specimens, guided navigation through predetermined 3D scenes, 3D animation, and interactive navigation through tri-planar sectional human cadaver images.

3.1 | Example 1: A simple, 3D-scanned anatomical specimen

A bitmap-textured, 3D-scanned axis vertebra was created that illustrates a simple, straightforward way of displaying a specimen in true-color quality within a 3D-PDF file (Figure 2A). The 3D viewport features the virtual specimen with only default navigation functions (zoom in/out, rotation, and translation) and no pre-configured interaction other than a default view and a single node in the *Model tree* control. The 3D-scanning approach presents limitations as to the morphological features that are captured and displayed, since this procedure only captures external morphology and thus fails to retain closed cavities or osseous canals. This procedure has in our experience provided most satisfactory results in digitizing bone specimens, however any prosected specimen devoid of anfractuous surfaces may be digitized to 3D-PDF files successfully.

3.2 | Example 2: A collection of individual meshes reconstructed from clinical imaging data

An assembly of skull bones reconstructed from CT scan files was produced (Figure 2B) to demonstrate how to generate complex or populated 3D scenes within a PDF file, based on multiple meshes reconstructed separately. The user can modify the scene as needed by hiding or showing any individual bone using the *Model tree* control. An advantage of reconstructing 3D models from CT data over surface scanning is that morphological features that are captured an displayed are not limited to the specimen's surface, but also internal structures are retained. In the present case, osseous canals and cavities such as ethmoidal cells, and maxillary and frontal sinuses, could be successfully captured and displayed in the 3D models. Other structures such as soft tissues, vasculature, or CNS structures are best segmented and reconstructed from plain or angiographic MRI data.

A limitation of reconstructing 3D models from clinical imaging data is that the extracted geometry data is devoid of color information, and thus the resulting surface meshes cannot be rendered in true-color quality as in the preceding example, but synthetic color must be assigned arbitrarily to each mesh by the document creator. In addition, transparency levels of reconstructed 3D models cannot be modified using the built-in rendering options in *Acrobat Reader*, but must be either set earlier at the U3D file production stage or accessed via *JavaScript* after PDF embedding.

3.3 | Example 3: Adding graphical and textual annotations

A 3D model of the temporal bone enriched with content annotations using 3D glyphs and built-in *3D comments* was created (Figure 3A). A variety of 3D glyphs can serve the purpose of pinpointing features of interest on the model or to hint towards properties or functions, for example, being the passage of specific vascular or nerve structures. Since these elements are added to the scene as polygon meshes at the U3D file creation stage, they will be rendered as regular surfaces within the 3D-PDF file, and the user will be able to hide and show them using the *Model tree* control. In order to clearly distinguish the 3D glyphs from the anatomical specimen, the former were all grouped here under a common category as *Markers* in the *Model tree*. On the other hand, the *3D comments* tool in *Acrobat Pro* provides the document creator with a mechanism to

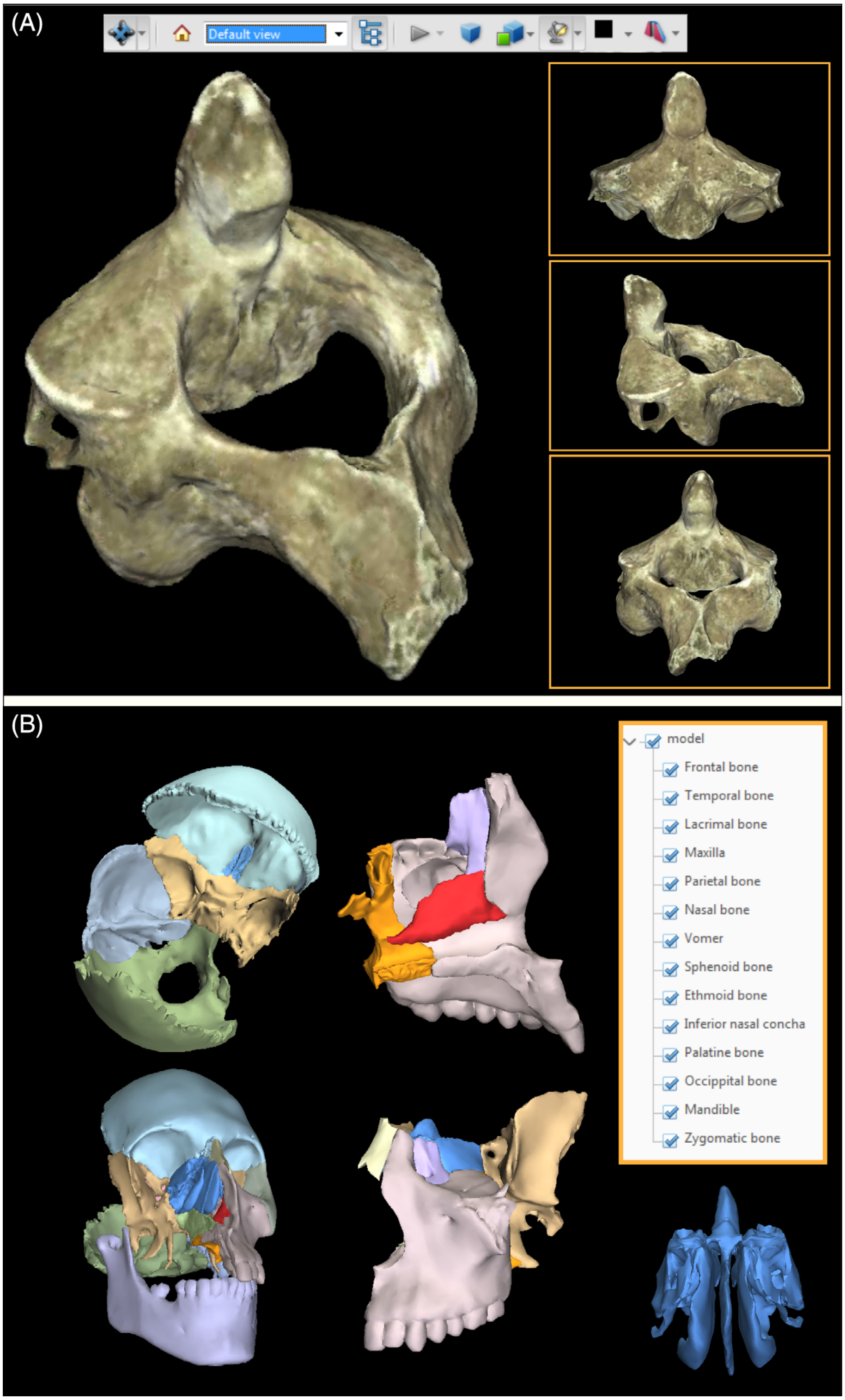

FIGURE 2 Unannotated representation of anatomical structures. (A) A 3D-scanned human axis is displayed with a photo-realistic appearance as a 3D-textured mesh (available as Data S1). The default scene (left side) and several additional views of the model (boxes on the right) are shown. User interaction is circumscribed to default options in *Acrobat Reader*, including model mobilization (rotation, translation, and zooming in/out) and manipulation of background color, lighting and rendering mode. (B) Partial assembly of a skull comprising multiple surface models of individual skull bones reconstructed from CT scan images of a Beauchêne preparation (available as Data S2). Three-dimensional polygon meshes representing individual bones are distinguished by unique synthetic colors. In addition to using the default navigation and visualization options, the user can hide and show any structure by selecting or deselecting the corresponding name entry on the adjacent *Model tree* control (box)

add textual information, as exemplified by numeric captions here, to any 3D model that has already been embedded in a PDF file. Three-dimensional comments are associated with pre-determined views by the document creator, and therefore they are replaced or removed automatically whenever the user switches views. This tool should be used judiciously, as an excess of captions can readily obscure vision of the models.

3.4 | Example 4: Using texture painting for highlighting surfaces of interest

This example of a 3D-PDF illustrates the use of texture painting as a mechanism to demonstrate areas of interest on a surface, namely the sites of muscle insertion on the mandible (Figure 3B). Highlighting surface areas using color codes, as accomplished here by texture painting,

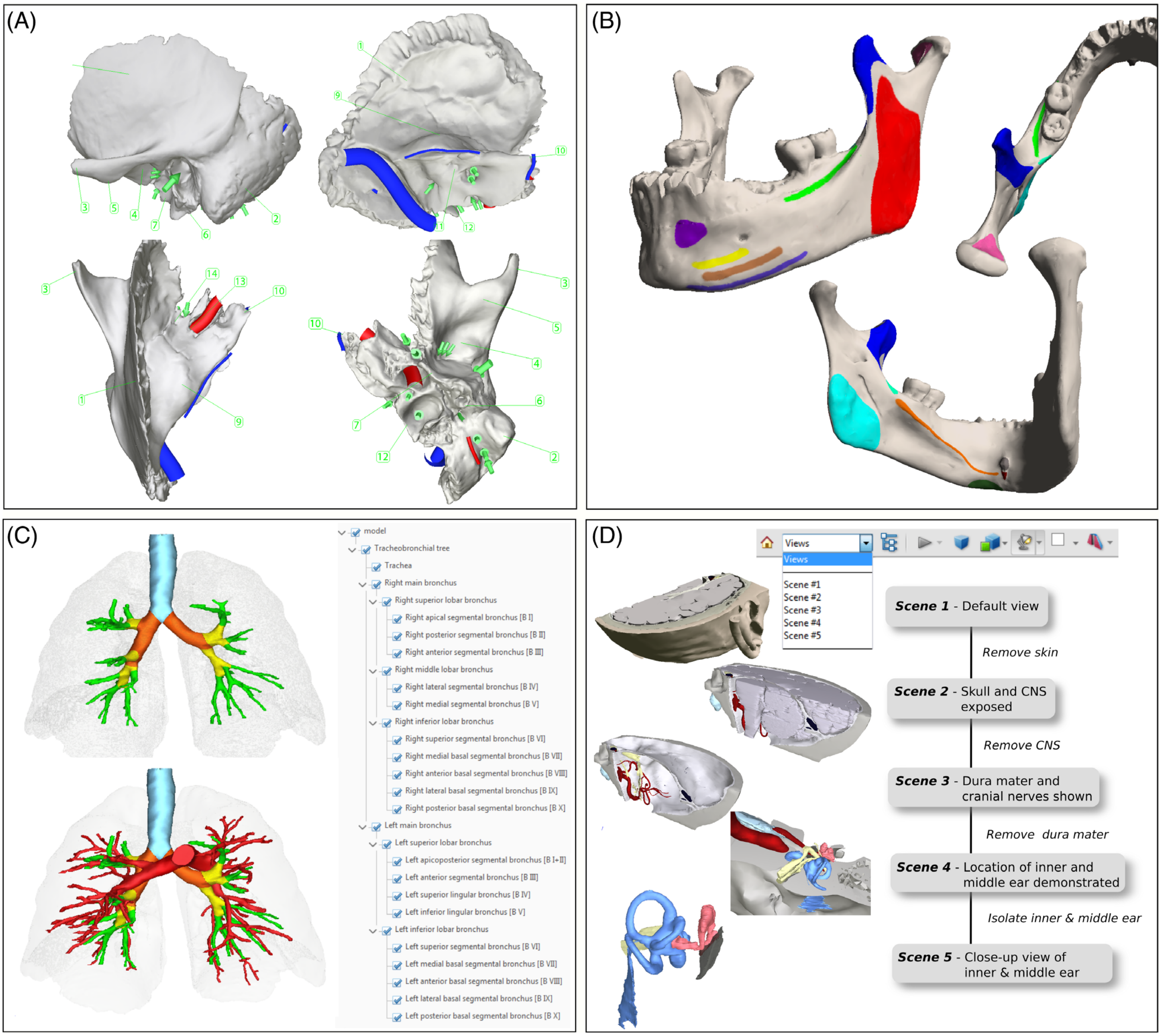

FIGURE 3 Enriching a 3D scene for learning purposes. (A) Annotation of a surface-scanned left temporal bone using 3D glyphs and built-in 3D comments (available as Data S3). Arrows and pipe-like meshes were added to the scene at the U3D file creation stage and thus prior to importation to *Acrobat Pro*, whereas 3D captions were later inserted using the *3D-comment* functionality in *Acrobat Pro*. Three-dimensional comments are replaced or removed by switching views, whereas 3D glyphs can be hidden or shown individually using the *Model tree* control. (B) Demarcation of regions of interest on a mandible surface model using bitmap-texture painting (available as Data S4). A 3D mesh generated by surface scanning is *wrapped* in a bitmap image on which the sites of muscle insertion are highlighted with unique colors. (C) Using the built-in *Model tree* control to describe hierarchy and provide structure information of the tracheobronchial tree (available as Data S5). Unique colors are used to distinguish bronchi as pertaining to main, lobar, and segmental categories, and the model structure specifies the hierarchical relationship of each bronchus to its parent and children. Each individual piece of bronchus is represented by a separate mesh and can be hidden or shown using the *Model tree* control (right side). Pulmonary arteries (lower model) and semi-transparent lungs were included for spatial reference. (D) A virtual dissection-like tour by navigating through sequentially arranged views (available as Data S6). Each predetermined view displays a particular subgroup of 3D structures representing a specific stage along a virtual dissection of the middle and inner ear. The flow diagram presents the sequence of five predetermined scenes, starting from viewing the specimen as a whole and reaching down to an isolated view of the middle and inner ear (shown on the left). The user can both proceed from any given stage to the next sequentially and switch scenes arbitrarily using the *Views* drop-down control in *Acrobat Reader* (top of image)

can effectively delimit and distinguish a region of interest, whenever the document creator wishes to point out a region or to parcellate a surface into discrete areas. Texture painting may in analogous manner become of use for mapping the distribution of territories on a surface, for example, those of cutaneous innervation or blood supply, or anatomic and functional regions on the cerebral cortex.

3.5 | Example 5: Hierarchy and structure information using the model tree

A 3D-PDF was created featuring the lungs and the constituent parts of the tracheobronchial tree, which also included information concerning structure and organizational categories of such parts in the *Model tree* control (Figure 3C). This example demonstrates how the document creator can deliver information regarding the organizational structure of an anatomical scene by pre-configuring the *Model tree* tool. In this manner, the end user can use the *Model tree* control in *Acrobat Reader* not only for hiding or displaying individual models in the scene, but also as a source of conceptual knowledge, in our present case regarding how each individual portion of the tracheobronchial tree should be recognized as pertaining to main, lobar or segmental categories, and their respective relations with parent or children bronchi.

Although the organizational hierarchy presented in the *Model tree* control corresponded to the structure of the tracheobronchial tree, the document creator can pre-configure this tool differently and 3D meshes representing an anatomic scene may well be grouped according to functional or topographic criteria, depending on the learning objectives. For instance, vascular and nerve supplies can be organized as separate branching trees or, alternatively, each neurovascular bundle or pedicle supplying a discrete region or organ can be considered as a group entity in the *Model tree* control.

3.6 | Example 6: Navigation through a predetermined sequence of 3D scenes

This example of a 3D-PDF file utilizes the *Views* control of *Acrobat Reader* to provide structure and direction to the user's activity (Figure 3D). The end user is intended to be guided through a virtual dissection just by switching from one predetermined view to the next using the *Views* drop-down menu. The user still keeps control over the visibility of scene components using the *Model tree* control, as well as over the view by rotating and displacing the models.

By arranging a sequence of 3D scenes, the document creator can suggest the background information required to best understand the core contents of the presentation in the succeeding scenes. This can be used by the instructor to organize contents during a classroom presentation, but also obviously by the student in a self-directed learning context. In addition, sequential 3D scenes can be used in the anatomy lab for assisting and guiding the student through the real dissection process. Despite the sense of direction that the described utilization of the *Views* tool creates, the end user does retain the ability to step back and revisit any scene, as well as to explore each scene by moving the camera around.

3.7 | Example 7: Viewing sectional anatomy interactively

An interactive 3D-PDF was created that features tri-planar sectional images of the anatomy of the human head. This example demonstrates the use of *JavaScript* scripting to enhance user interaction with the embedded 3D contents (Figure 4A). *JavaScript* was used here to link the action of clicking on a button to the effect of showing the previous or the next section in a particular axis while hiding all the remaining sections, thus creating the effect of a navigation through sectional images. Nevertheless, the *Model tree* control provides flexibility to hide or show any other cross-section any time.

Cross-sectional anatomy is a difficult subject for students to understand due to the often intricate topographic relationships among structures. An interactive tri-planar view helps the student analyze a cross-section in any plane and mobilize it to a convenient viewpoint, while keeping spatial references with the images in the other two orthogonal planes. Although the present example was based solely on cadaver cross-sections, an analogous graphical strategy may be used for presenting MRI or CT imaging data or combined, registered images of different modalities, for example, cadaver images correlated with clinical imaging data.

3.8 | Example 8: 3D animation

An interactive 3D-PDF was produced in which the user can play a 3D animation of the rotational and translational movements of the mandibular condyles. This example illustrates a relatively simple procedure to generate 3D movement in a PDF file using *JavaScript* (Figure 4B). Animation in a 3D-PDF file can be managed in two different ways. The U3D file format supports animation, and thus 3D models can be pre-configured as animated content at the U3D file stage, that is, prior to PDF embedding. This method has been used previously to simulate deterministic movements of a hip joint of an insect (van de Kamp et al., 2014). Most conveniently, *JavaScript* scripting as used in the present example allows the user to play the animation by clicking on a button, and provides precise control to the document creator to establish a number of parameters of an animation, including speed and smoothness. Animation can be triggered by the user irrespective of the camera position and keeps going during zooming in/out within the scene or during rotation or panning of the 3D model.

4 | DISCUSSION

The present report illustrates a range of technical possibilities that the 3D-PDF technology offers for delivering anatomical information and knowledge. This is a tool of great potential for the teaching and learning of anatomy. The proof-of-concept examples provided show that 3D content embedded in PDF files is not generally inferior to dedicated multimedia software applications in terms of quality, while exhibiting high flexibility, convenience and cross-platform compatibility, as well as a favorable learning curve both for the document creator and the user.

4.1 | Quality as a learning resource

Recent decades have witnessed an explosion of new digital learning materials and resources, including interactive 3D models and software

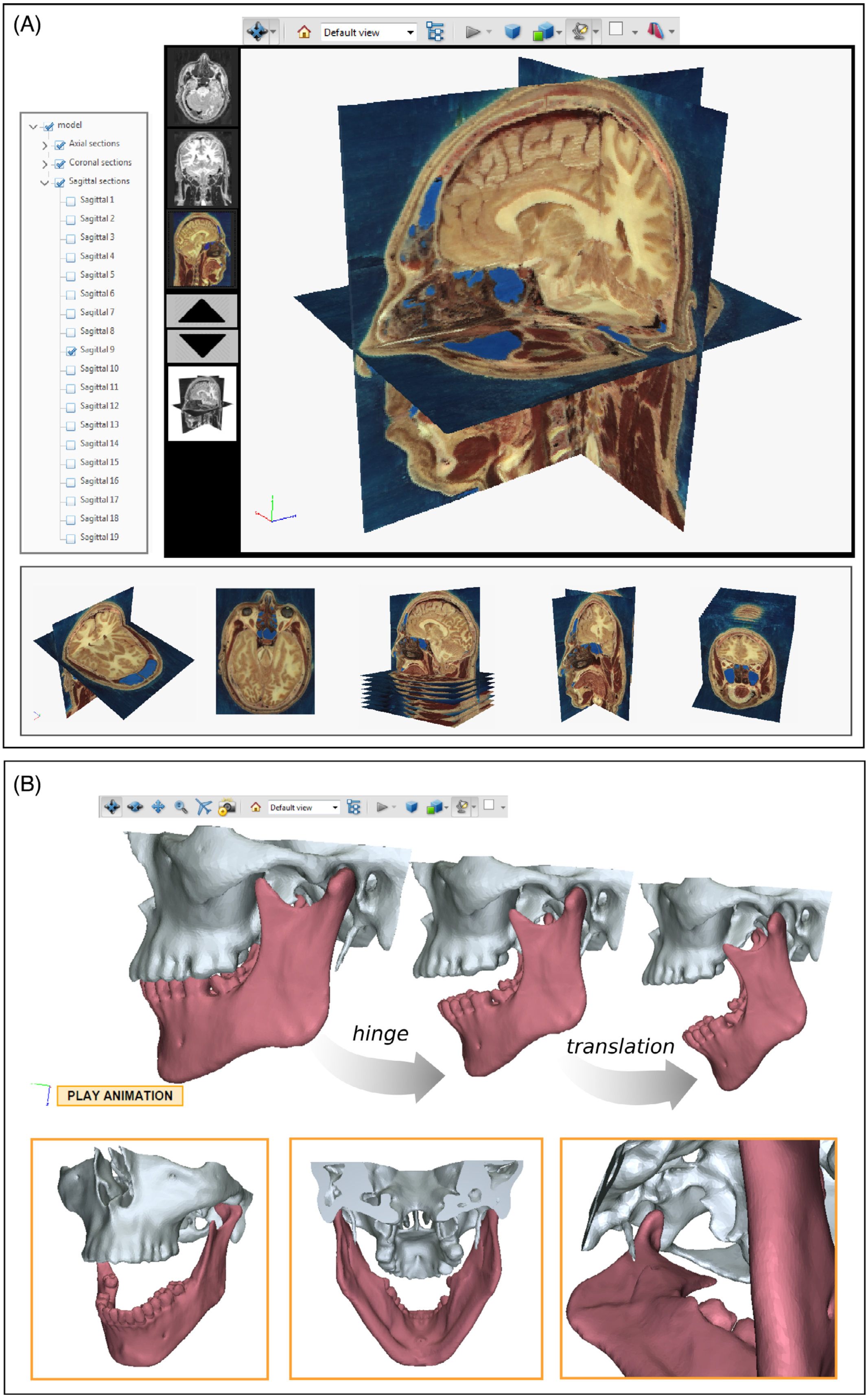

FIGURE 4 Legend on next page.

applications for visualizing human anatomy. There is now substantial evidence that 3D computer visualization of digital anatomical models is an effective learning tool in terms of both factual and spatial knowledge (Murgitroyd et al., 2015; Triepels et al., 2020; Yammine & Violato, 2015). In a recent systematic search, Zilverschoon et al. (2019) reviewed 20 different electronic applications for anatomy learning, both commercially and noncommercially available, in order to compare features of computer-based instructional systems. Model realism, which was a primary measure in their study, was assigned the highest score when an anatomical 3D model was as realistic in both shapes and visual details as possible. Arguably, 3D-PDF files attain the highest quality standards in this respect, as exemplified by the photo-realistic appearance of textured, surface-scanned specimens (*Example 1*). Despite the obvious usefulness of the PDF for presenting surface-scanned, bitmap-textured 3D graphics for education purposes, this technology has to the best of the author's knowledge not been presented in literature before, in the educational context. Also novel in the educational context is the present approach of utilizing texture painting for demarcating territories on a specimen surface (*Example 4*), in contrast with 3D scenes thus far presented in the literature, which commonly feature plain, nontextured surface models (Kato et al., 2014, 2016; Chung, Chung, & Park, 2015; Chung et al., 2017, 2018; Kim et al., 2013, Kim et al., 2015; Park et al., 2015; Shin, Kwon, et al., 2015; Wu et al., 2017). Enriching a 3D model or scene with additional information can also be accomplished as shown here either using the built-in *3D comment* tool in Acrobat, much like any two-dimensional illustration is annotated, or by adding 3D glyphs in the appropriate locations. In addition, it is shown that rotation, displacement, zooming in/out, the presence of clickable structures, changeable transparency, and availability of anatomical descriptions, which all are considered as quality indicators of program functionality (Zilverschoon et al., 2019), are fully supported by the 3D-PDF. Moreover, Zilverschoon et al. (Zilverschoon et al., 2019) regarded virtual dissection, defined as dragging or clicking away items to expose underlying structures, as an advanced functionality for studying anatomy. It is shown here that managing visibility of rendered items is trivial in 3D-PDF files using the *Model tree* control. This tool allows to group and visualize 3D models by systems (e.g., skeletal, muscular, cardiovascular, etc.; Shin, Chung, et al., 2012; Kim et al., 2015), as well as to convey information regarding nomenclature and structural hierarchy (*Examples 2*, *5*). Furthermore, the document author can create a dissection-like experience by pre-configuring sequential scenes with an orderly removal of anatomical structures using the *Manage views* functionality (*Example 6*).

4.2 | Flexibility and adaptability

Portable Document Format files with embedded 3D content exhibit unprecedented flexibility at a variety of levels. Granularity of contents, level of detail in both geometry and textures, and complexity of 3D scenes can all easily be adapted to learning objectives. In addition, appropriately designed 3D-PDF files could serve to integrate 3D visualizations and learning object technologies. Learning objects are a specialized type of educational software, where content is broken down into compact, self-contained chunks that can be repurposed for different educational applications (Churchill, 2007; Wiley, 2002). A learning object includes metadata, that is, data about the data it contains, in order to support discoverability and reusability. In PDF files, basic metadata are added through file info and document properties, and almost every desired information can be included in PDF files as metadata attribute using the so-called *Extensible Metadata Platform* (XMP), a more powerful XML format-based metadata standard created by *Adobe* (Adobe Systems Inc, 2012a, 2012b, 2012c).

In addition, using PDF files to present interactive 3D imagery lends itself to integration into a variety of instructional design and learning styles. It only seems intuitive to embed 3D illustrations instead of 2D graphics in text-based, guided or primarily expositive learning materials. Arguably, 3D-PDF files compare advantageously to sophisticated augmented- and virtual reality frameworks as to the ability to combine text and 3D graphics, as the latter both seem to be less then ideally suited to present textual content. Indeed, the adoption of 3D graphics within PDF files has been advocated as a desirable improvement in scientific reports (Barnes et al., 2013; Danz & Katsaros, 2011; Murienne et al., 2008; Newe & Becker, 2018; Ruthensteiner & Hess, 2008). At the other end of the continuum of learning styles, a virtual specimen can be presented *as is* in a 3D-PDF file, with minimal or no labeling, much in the way that a real specimen can be used for active exploration in the anatomy lab, in more of a self-directed learning style. Importantly, 3D-PDF files also can represent sectional anatomy interactively, which also renders this digital resource particularly useful for case-based, clinically oriented modules.

FIGURE 4 Enhancing user interaction with *JavaScript* scripting. (A) Sectional anatomy viewer endowed with a simple, *JavaScript*-powered user interface (available as Data S7). Buttons allow the user to navigate from one section to the next in either direction, as well as to reset the scene to a default tri-planar view. The *Model tree* control (box on left margin) lists all sectional images available for viewing, and allows the user to hide or show individual or multiple images. Examples of various views obtained by operating the *Model tree* control are shown at the bottom.
(B) Simple animation of a 3D anatomical scene presented within a PDF file (available as Data S8). The spatial coordinates of a polygon mesh representing the human mandible in U3D format are manipulated using *JavaScript* in order to recreate rotation and translation movements of the mandibular condyle from the centric position (top of figure). Annotated sample *JavaScript* code controlling basic animation of a 3D model in U3D format is provided in Appendix II. The user can trigger animation by clicking on a custom-created, *JavaScript*-powered button, while keeping control on navigation parameters and visualization of the 3D scene. Additional anterolateral, posterior, and medial views, that is, views other than the default one, are shown in boxes at the bottom

The contribution of *JavaScript* scripting to enhancing functionality and user interaction in 3D-PDF files deserves special mention. Although the present report provides only two examples of *JavaScript*-enabled buttons (*Examples 7* and *8*), the *JavaScript for Acrobat* specification includes a variety of additional control elements, such as text fields, drop-down menus, list boxes, etc, that the document creator can incorporate to link user input to specific preconfigured actions over the 3D content. This extended functionality renders PDF files virtually fully interactive and practically comparable to dedicated software applications. Examples for application-like PDF files can be found in Reina et al. (2016), Valera-Melé et al. (2018) and Newe, Becker, and Schenk (2014). Here, *JavaScript* was also used to control the visibility of sectional images (*Example 7*) and thus to control navigation from one sectional image to the next. A similar approach was used by Reina et al. (2016) and Valera-Melé et al. (2018). This provides an alternative to the *Model tree* control to manage the visibility of sections, which has been most widely used in previous reports (Chung, Chung, & Park, 2015; Jang et al., 2015; Shin et al., 2018a, 2018b; Shin, Chung, et al., 2012; Shin, Jang, et al., 2012; Shin, Kwon, et al., 2015; Shin, Lee, et al., 2015). An additional, important functionality brought about by *JavaScript* scripting is the ability to generate and control 3D animation. As opposed to movies, which provide no user-interactivity other than stop-and-go and, in some cases, also forward/backward control, animation of 3D contents within a PDF file offers the opportunity to combine motion information with full user interactivity, including the ability to manipulate views and visibility of scene elements at any time during the 3D motion. The interest of 3D animation to illustrate human functional morphology seems obvious, for example, for contents with a focus on biomechanics of neuromuscular function or visceral motion. Such applications may also benefit from future developments including 4D morphing animations as suggested by other authors (Ruthensteiner & Hess, 2008).

4.3 | Towards adopting 3D-PDF technology

For PDF-embedded 3D graphics to gain widespread use as a teaching and learning resource, this technology must be convenient to use and perceived as useful both by the document creator and the student. PDF files can be viewed in most modern digital devices, irrespective of the software, hardware, and operating system. Students can thus carry around a wide range of learning resources on a single device, including interactive 3D imagery, without the need to install dedicated software applications, and can revisit not only textual content but also learning materials that would otherwise be circumscribed to the gross anatomy lab, such as digitized specimens or clinical imaging data.

In addition, however, 3D-PDF files should also be valued positively as a learning tool by the student. In this respect, the technology acceptance model (TAM; Davis, 1989; Davis, Bagozzi, & Warshaw, 1989) is a widely applied and empirically tested theoretical model to explain end-users' acceptance behavior. This model is one of the most influential extensions of Ajzen and Fishbein's (1980) *Theory of Reasoned Action*, which postulates that beliefs influence attitudes and that these in turn lead to intentions that ultimately generate behavior. Future studies are needed to investigate *perceived ease of use* and *perceived usefulness* of PDF-embedded 3D content, as both are considered by the TAM as critical factors influencing behavioral intention to use.

From the document creator's perspective, a limitation of the 3D-PDF file generation process is the number of steps involved. It is noteworthy, however, that most of the effort is actually dedicated to creating the 3D models. If starting from segmentation of cross-sectional data, the process of generation of 3D models requires using several software applications and may be time consuming. By contrast, 3D models are readily obtained by surface scanning. On the other hand, the present report shows that converting standard 3D files into U3D format and embedding this content in a PDF file is accomplished easily using both commercially available or freely available software tools. Ongoing efforts to simplify the U3D file generation pipeline (Newe, 2015, 2016; Newe & Ganslandt, 2013) will probably contribute to facilitating the process of production of 3D-PDF files. The design and production of 3D-PDF files with more complex and sophisticated user interactions will conceivably require greater effort on the part of the document creator, in particular whenever *JavaScript* scripting is used to implement user interaction. It is important to note, however, that presentation of 3D content in PDF will take considerably less effort than developing dedicated 3D visualization software, while still providing great flexibility and functionality.

5 | CONCLUSIONS

The present report shows a variety of procedures to generate and display 3D anatomical scenes embedded in PDF, demonstrating that 3D-PDF files are generally comparable to multimedia and dedicated 3D visualization software in terms of quality and flexibility. Importantly, 3D-PDF files can be read in most modern digital devices. This emerging technology is therefore convenient and best suited to be adopted in the academic scenario for communication and delivery of anatomical information and knowledge, and particularly in the context of teaching and learning human anatomy.

ORCID

Jon Jatsu Azkue https://orcid.org/0000-0002-8609-5500

REFERENCES

Ackerman, M. J. (1998). The visible human project. *Proceedings of the IEEE*, *86*(3), 504–511.

Adobe Systems, Inc. Adobe portable document format, Version 1.6. 2004. https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDFReference16.pdf.

Adobe Systems, Inc. Adobe supplement to the ISO 32000. 2008. https://www.adobe.com/content/dam/acom/en/devnet/pdf/adobe_supplement_iso32000.pdf.

Adobe Systems, Inc. XMP specification Part 1. 2012a. Data model, serialization, and core properties. https://wwwimages2.adobe.com/content/dam/acom/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart1.pdf.

Adobe Systems, Inc. XMP specification Part 2. Additional properties 2012b. https://wwwimages2.adobe.com/content/dam/acom/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2014-12/XMPSpecificationPart2.pdf.

Adobe Systems, Inc. XMP specification Part 3. Storage in files. 2012c. https://wwwimages2.adobe.com/content/dam/acom/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart3.pdf.

Adobe Systems, Inc. *JavaScript* for Acrobat 3D annotations API reference. 2015. https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/AcrobatDC_js_3d_api_reference.pdf.

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Englewood Cliffs.

Aziz, M. A., McKenzie, J. C., Wilson, J. S., Cowie, R. J., Ayeni, S. A., & Dunn, B. K. (2002). The human cadaver in the age of biomedical informatics. *The Anatomical Record*, *269*(1), 20–32.

Barnes, D. G., Vidiassov, M., Ruthensteiner, B., Fluke, C. J., Quayle, M. R., & McHenry, C. R. (2013). Embedding and publishing interactive, 3-dimensional, scientific figures in portable document format (PDF) files. *PLoS One*, *8*(9), e69446.

Chapman, A. D., Turland, N. J., & Watson, M. F. (2010). Report of the special committee on electronic publication. *Taxon*, *59*(6), 1853–1862.

Chung, B. S., Chung, M. S., & Park, J. S. (2015). Six walls of the cavernous sinus identified by sectioned images and three-dimensional models: Anatomic report. *World Neurosurgery*, *84*(2), 337–344.

Chung, B. S., Kwon, K., Shin, B. S., & Chung, M. S. (2017). Surface models and gradually peeled volume model to explore hand structures. *Annals of Anatomy*, *211*, 202–206.

Chung, B. S., Chung, M. S., Shin, B. S., & Kwon, K. (2018). Three software tools for viewing sectional planes, volume models, and surface models of a cadaver hand. *Journal of Korean Medical Science*, *33*(8), e64.

Churchill, D. (2007). Towards a useful classification of learning objects. *Educational Technology Research and Development*, *55*(4), 479–497.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2008). MeshLab: An open-source mesh processing tool. *Sixth Eurographics Italian Chapter Conference*, *2008*, 129–136.

Danz, J. C., & Katsaros, C. (2011). Three-dimensional portable document format: A simple way to present 3-dimensional data in an electronic publication. *American Journal of Orthodontics and Dentofacial Orthopedics*, *140*(2), 274–276.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, *35*(8), 982–1003.

Federative Committee on Anatomical Terminology. (1998). *Terminologia Anatomica*. Germany, Thieme Verlag.

Hitchcock, S., Carr, L., & Hall, W. (1997). Web journals publishing: A UKperspective. *Serials*, *10*(3), 285–299.

Jang, H. G., Chung, M. S., & Shin, D. S. (2015). Portable document format file containing the surface models to learn the stereoscopic shape of foot muscles. *International Journal of Morphology*, *33*(4), 1287–1292.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*, 321–331.

Kato, A., Ziegler, A., Higuchi, N., Nakata, K., Nakamura, H., & Ohno, N. (2014). Aetiology, incidence and morphology of the C-shaped root canal system and its impact on clinical endodontics. *International Endodontic Journal*, *47*, 1012–1033.

Kato, A., Ziegler, A., Utsumi, M., Ohno, K., & Takeichi, T. (2016). Three-dimensional imaging of internal tooth structures: Applications in dental education. *Journal of Oral Biosciences*, *58*(3), 100–111.

Kim, B. C., Chung, M. S., Park, H. S., Shin, D. S., & Park, J. S. (2015). Accessible and informative sectioned images and surface models of the maxillofacial area for orthognathic surgery. *Folia Morphologica*, *74*(3), 346–351.

Longhurst, G. J., Stone, D. M., Dulohery, K., Scully, D., Campbell, T., & Smith, C. F. (2020). Strength, weakness, opportunity, threat (SWOT) analysis of the adaptations to anatomical education in the United Kingdom and Republic of Ireland in response to the Covid-19 pandemic. *Anatomical Sciences Education*, *13*(3), 301–311.

McLachlan, J. C., Bligh, J., Bradley, P., & Searle, J. (2004). Teaching anatomy without cadavers. *Medical Education*, *38*, 418–424.

Murgitroyd, E., Madurska, M., Gonzalez, J., & Watson, A. (2015). 3D digital anatomy modelling: Practical or pretty? *The Surgeon*, *13*(3), 177–180.

Murienne, J., Ziegler, A., & Ruthensteiner, B. (2008). A 3D revolution in communicating science. *Nature*, *453*(7194), 450.

Newe, A. (2015). Towards an easier creation of three-dimensional data for embedding into scholarly 3D PDF (portable document format) files. *PeerJ*, *3*, e794.

Newe, A. (2016). Enriching scientific publications with interactive 3D PDF: An integrated toolbox for creating ready-to-publish figures. *PeerJ Computer Science*, *2*, e64.

Newe, A., & Becker, L. (2018). Three-dimensional portable document format (3D PDF) in clinical communication and biomedical sciences: Systematic review of applications, tools, and protocols. *JMIR Medical Informatics*, *6*(3), e10295.

Newe, A., & Ganslandt, T. (2013). Simplified generation of biomedical 3D surface model data for embedding into 3D portable document format (PDF) files for publication and education. *PLoS One*, *8*(11), e79004.

Newe, A., Becker, L., & Schenk, A. (2014). Application and evaluation of interactive 3D PDF for presenting and sharing planning results for liver surgery in clinical routine. *PLoS One*, *9*(12), e115697.

Notzer, N., Zisenwine, D., Oz, L., & Rak, Y. (2006). Overcoming the tension between scientific and religious views in teaching anatomical dissection: The Israeli experience. *Clinical Anatomy*, *19*(5), 442–447.

Older, J. (2004). Anatomy: A must for teaching the next generation. *The Surgeon*, *2*(2), 79–90.

Park, H. S., Chung, M. S., Shin, D. S., Jung, Y. W., & Park, J. S. (2015). Whole courses of the oculomotor, trochlear, and abducens nerves, identified in sectioned images and surface models. *Anatomical Record (Hoboken)*, *298*, 436–443.

Raja, D. S., & Sultana, B. (2012). Potential health hazards for students exposed to formaldehyde in the gross anatomy laboratory. *Journal of Environmental Health*, *74*(6), 36–40.

Reina, M. A., Lirk, P., Puigdellívol-Sánchez, A., Mavar, M., & Prats-Galino, A. (2016). Human lumbar ligamentum flavum anatomy for epidural anesthesia: Reviewing a 3D MR-based interactive model and postmortem samples. *Anesthesia and Analgesia*, *122*(3), 903–907.

Ruthensteiner, B., & Hess, M. (2008). Embedding 3D models of biological specimens in PDF publications. *Microscopy Research and Technique*, *71*(11), 778–786.

Shin, D. S., Chung, M. S., Park, J. S., Park, H. S., Lee, S., Moon, Y. L., & Jang, H. G. (2012). Portable document format file showing the surface models of cadaver whole body. *Journal of Korean Medical Science*, *27*(8), 849–856.

Shin, D. S., Jang, H. G., Park, J. S., Park, H. S., Lee, S., & Chung, M. S. (2012). Accessible and informative sectioned images and surface models of a cadaver head. *The Journal of Craniofacial Surgery*, *23*(4), 1176–1180.

Shin, D. S., Kwon, K., Shin, B.-S., Park, H. S., Lee, S., Lee, S.-B., & Chung, M. S. (2015). Surface models and gradually stripped volume model to explore the foot muscles. *Journal of Experimental and Clinical Anatomy*, *9*(1), 19–25.

Shin, D. S., Lee, S., Park, H. S., Lee, S. B., & Chung, M. S. (2015). Segmentation and surface reconstruction of a cadaver heart on mimics software. *Folia Morphologica*, *74*(3), 372–377.

Shin, D. S., Shim, Y. J., & Kim, B. C. (2018a). Sectioned images and 3D models of a cadaver head with reference to dermal filler injection. *Annals of Anatomy*, *217*, 34–39.

Shin, D. S., Shim, Y. J., & Kim, B. C. (2018b). Sectioned images and surface models of a cadaver head with reference to botulinum neurotoxin injection. *Folia Morphologica*, *77*(3), 564–569.

Spitzer, V., Ackerman, M. J., Scherzinger, A. L., & Whitlock, D. (1996). The visible human male: A technical report. *Journal of the American Medical Informatics Association*, *3*(2), 118–130.

Sugand, K., Abrahams, P., & Khurana, A. (2010). The anatomy of anatomy: A review for its modernization. *Anatomical Sciences Education*, *3*(2), 83–93.

Tabernero Rico, R. D., Juanes Méndez, J. A., & Prats-Galino, A. (2019). Application of PDF software with 3D functionalities in radiological models of the Skull Base: Characteristics, experience and solutions. *Journal of Medical Systems*, *43*, 103.

Triepels, C. P. R., Smeets, C. F. A., Notten, K. J. B., Kruitwagen, R. F. P. M., Futterer, J. J., Vergeldt, T. F. M., & Van Kuijk, S. M. J. (2020). Does three-dimensional anatomy improve student understanding? *Clinical Anatomy*, *33*(1), 25–33.

Valera-Melé, M., Puigdellívol-Sánchez, A., Mavar-Haramija, M., Juanes-Méndez, J. A., San-Román, L., de Notaris, M., & Prats-Galino, A. (2018). A novel and freely available interactive 3D model of the internal carotid artery. *Journal of Medical Systems*, *42*(4), 72.

van de Kamp, T., dos Santos Rolo, T., Vagovič, P., Baumbach, T., & Riedel, A. (2014). Three-dimensional reconstructions come to life–interactive 3D PDF animations in functional morphology. *PLoS One*, *9*(7), e102355.

Wang, H., Merchant, S. N., & Sorensen, M. S. (2007). A downloadable three-dimensional virtual model of the visible ear. *ORL: Journal for Oto-rhinolaryngology and Its Related Specialties*, *69*(2), 63–67.

Wiley, D. A. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The instructional use of learning objects. Bloomington, Association for Educational Communications and Technology* (pp. 1–23). Agency for Instructional Technology Association for Educational Communications & Technology.

Wu, Y., Dabhoiwala, N. F., Hagoort, J., Tan, L. W., Zhang, S. X., & Lamers, W. H. (2017). Architectural differences in the anterior and middle compartments of the pelvic floor of young-adult and postmenopausal females. *Journal of Anatomy*, *230*(5), 651–663.

Yammine, K., & Violato, C. (2015). A meta-analysis of the educational effectiveness of three-dimensional visualization technologies in teaching anatomy. *Anatomical Sciences Education*, *8*(6), 525–538.

Yushkevich, P. A., Piven, J., Hazlett, H. C., Smith, R. G., Ho, S., Gee, J. C., & Gerig, G. (2006). User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, *31*(3), 1116–1128.

Zilverschoon, M., Kotte, E. M. G., van Esch, B., Ten Cate, O., Custers, E. J., & Bleys, R. L. A. W. (2019). Comparing the critical features of e-applications for three-dimensional anatomy education. *Annals of Anatomy*, *222*, 28–39.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Azkue, J. J. (2021). Embedding interactive, three-dimensional content in portable document format to deliver gross anatomy information and knowledge. *Clinical Anatomy*, *34*(6), 919–933. https://doi.org/10.1002/ca.23755

APPENDIX I

Example *LaTex* code using the *Movie 15* (**A**) or *Media 9* (**B**) packages to generate a 3D-PDF with an embedded U3D file named *cube.u3d*. The script in **A** is generated as a TEX file automatically by *MeshLab* software upon exporting a 3D mesh as a U3D file. Please note that although *Movie 15* package is deprecated and superseded by the *Media 9* package, the latest update of *MeshLab* (v2020.12d, built December 2020) still makes use of the former. An alternative version of the code adapted to the *Media 9* package is provided in **B**.

A. *LaTeX code generated by MeshLab software to use with* Movie 15 *package*

```
\documentclass[a4paper]{article}
\usepackage[3D]{movie15}
\usepackage{hyperref}
\usepackage[UKenglish]{babel}
\begin{document}
\includemovie[
     poster,
     toolbar, %same as 'controls'
     label=cube.u3d,
     text=(cube.u3d),
     3Daac=60.000000, 3Droll=0.000000, 3Dc2c=-0.000000 -3.464000 0.000000, 3Droo=3.464000, 3Dcoo=-0.000000 0.000000 -0.000000,
     3Dlights=CAD,
]{\linewidth}{\linewidth}{cube.u3d}
\end{document}
```

B. *LaTeX code adapted to use with* Media 9 *package*

```
\documentclass[a4paper]{article}
\usepackage{media9}
\usepackage{hyperref}
\usepackage[UKenglish]{babel}
\usepackage{graphicx}
\usepackage{xspace}
\begin{document}
     \includemedia[
     label=cube.u3d,
     3Daac=60.000000, 3Droll=0.000000, 3Dc2c=-0.000000 -3.464000 0.000000, 3Droo=3.464000, 3Dcoo=-0.000000 0.000000 -0.000000,
     3Dlights=CAD,
     ]{}{cube.u3d}
\end{document}
```

APPENDIX II

Example *JavaScript* code for simple animation of a 3D mesh (named *cube*) within a U3D file. *JavaScript* provides control over the execution timeline of *rotation* and *translation* methods, allowing the document creator to dynamically change the spatial coordinates of the 3D object. Code highlighted in blue represents parameters that can be modified to determine the extent, speed, and smoothness of motion. For simplicity, rotation is executed only in the X-axis.

***JavaScript* code**	**Comment**
rotation= 360;	Final angle of rotation (around the X-axis).
translationX = 2.5;	Final translation along the X-axis.
translationY = 8.5;	Final translation along the Y-axis.
translationZ = 3.0;	Final translation along the Z-axis.
timeInterval = 10;	Time interval between frames, in milliseconds.
currentFrame = 0;	A counter to control the progress of animation.
c3d = this.getAnnots3D(0)[0].context3D;	Access to the first 3D object on the first page of the PDF file.
numberOfFrames = 200;	Number of *frames* comprising the animation (fractions or subdivisions of the movement); the higher the number of frames, the smoother the sensation of movement.

if (typeof c3d != "undefined") {
id = app.setInterval("animate();", timeInterval);
}
	Checks if the 3D object has been successfully accessed. If so, a function is called to execute a fraction of the motion. After waiting for a predetermined time interval, the function is called again for executing the next fraction of the motion.

function animate() {	This block executes the actual movements.

if (step == numberOfFrames) {
app.clearInterval(id);
}
	Stops animation if the last frame has been reached.

cube = c3d.scene.meshes.getByName("cube");	Provides access to the *cube* mesh in the 3D scene.
cube.transform.rotateAboutXInPlace((Math.PI * rotation)/(180 * numberOfFrames));	Rotates the mesh by a fraction of the predetermined angle.
cube.transform.translateInPlace(new c3d.Vector3(translationX/numberOfFrames, translationY/numberOfFrames, translationZ/numberOfFrames));	Displaces the cube along each axis by a fraction of the predetermined displacement extent.
currentFrame = currentFrame + 1;	Increases the counter by one, to push to the next frame.
}	