The LiverAnatomyExplorer

A WebGL-Based Surgical Teaching Tool

Steven Birr, Jeanette Mönch, and Dirk Sommerfeld ▪ *University of Magdeburg*

Uta Preim ▪ *Clinical Center Magdeburg*

Bernhard Preim ▪ *University of Magdeburg*

The LiverAnatomyExplorer combines traditional clinical 2D imagery with interactive Web-based 3D models derived from patient-specific image data. The tool is enhanced by surgical videos, a self-assessment tool, and an online authoring tool with which instructors can manage the presented case studies and create multiple-choice quizzes.

Traditionally, medical students learn basic disease-related anatomical facts and therapy options through textbooks, lectures, and seminars. Training in surgical skills—for example, suturing—occurs in special courses with small groups of students. The students are supplied with appropriate resources such as slide shows, journals, and textbooks. They must apply the knowledge gained from these materials to real clinical situations.

To supplement these methods, researchers are increasingly developing Web-based learning systems.[1] With these systems, students can review lectures and workshops anytime and anywhere. Instructors can enhance lectures with multimedia or online resources. Furthermore, they can quickly change the online course materials. Web-based learning systems can be accessed, reused, and studied even long after the course ends.[2]

However, few Web-based medical learning systems offer interactive 3D models or sufficient user feedback. To help fill this void, we developed the LiverAnatomyExplorer, an anatomical and surgical teaching tool.[3] It employs three state-of-the-art Web technologies: SVG (Scalable Vector Graphics) for displaying 2D vector data, X3D (Extensible 3D) for medical 3D visualizations, and HTML5 and WebGL (Web Graphics Library) for the rendering engine. (For more on these technologies, see the related sidebar.) It includes patient-specific 2D image data with colored overlays, easy-to-use interactive polygonal 3D models, diagnosis and surgical reports, surgical videos, and self-assessment exercises.

This article extends our previous research by presenting more implementation details and describing an authoring tool for managing course material and creating quizzes. We also further discuss and clarify evaluation results.

Medical Learning Systems and 3D Visualizations

Medical learning systems are helpful for surgeons in their first years of training. In particular, they benefit from 3D visualizations of human anatomy and diseases. The most obvious benefits of medical 3D models are the efficient overview of anatomical situations and the improved interpretation of complex spatial relationships between organs, tumors, and at-risk structures such as blood vessels.

Several offline systems teach anatomical basics by employing 3D models for interactive visualizations—for example, Voxel-Man 3D-Navigator: Inner Organs (www.voxel-man.de/3d-navigator/inner_organs) and PrimalPictures 3D Human Anatomy (www.primalpictures.com). Simulators such as Voxel-Man Dental (www.voxel-man.de/simulator/dental) improve training of surgical procedures by providing specialized work spaces and skill assessment tools.

Beyond that, there are the few Web-based surgical learning systems that Carolina Pape-Köhler and her colleagues reviewed—for example, WebSurg (www.websurg.com) and SurgyTec (www.surgytech.com).[4] None of these systems fulfilled the reviewers'

 Published by the IEEE Computer Society

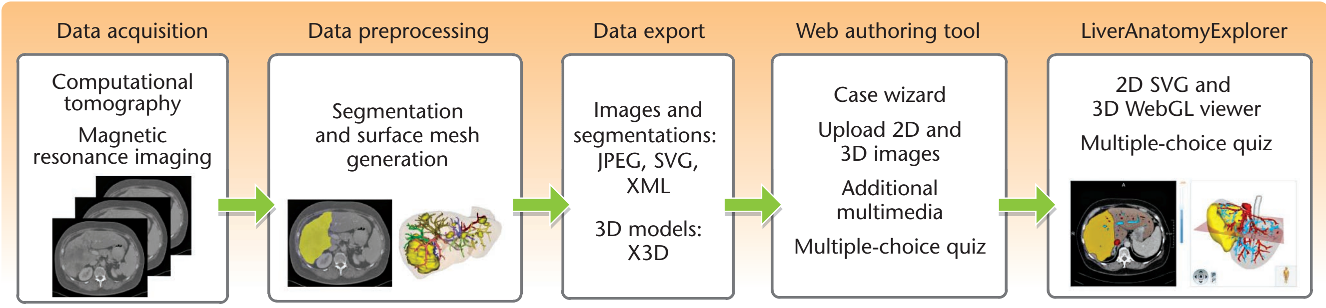


Figure 1. The LiverAnatomyExplorer pipeline. Data acquisition acquires clinical image data of the liver. Data preprocessing segments the liver, vascular structures, and tumors and generates surface meshes, using an external application such as MeVisLab. The image stack and segmentation objects are exported as JPEG and SVG (Scalable Vector Graphics) files; 3D models are exported as X3D (Extensible 3D) files. The authoring tool integrates the exported data and additional multimedia content into the Web platform. Students use the synchronized 2D and 3D viewers and the self-assessment tool to learn liver anatomy. WebGL stands for Web Graphics Library.

criteria regarding currency, scientific content, and validity. Most of them provide only videos of surgical interventions and limited Flash-based medical animations, instead of interactive patient-specific 3D visualizations. Self-assessment tools that provide immediate feedback are largely missing. These inadequacies led us to develop the LiverAnatomyExplorer.

Requirements Analysis

Before implementing the LiverAnatomyExplorer, we asked 176 medical students and 19 clinicians to complete an online survey. We wanted to determine their computer and 3D skills, the typical learning materials, and user needs. All the students and 13 clinicians completed the survey.

Most of the students (125, or 71 percent) and clinicians (9, or 69 percent) used their computer daily. However, 97 percent (171) of the students and 54 percent (7) of the clinicians had rarely or never used 3D visualizations. Obviously, we were targeting people who were unfamiliar with interactive 3D graphics. So, we decided to provide simple interaction methods for exploring 3D models.

Regarding user needs, many students wanted clinical-case collections with medical imagery and 3D representations of anatomical basics. Furthermore, they wanted to use these Web-based tools to prepare for examinations and to test their knowledge. So, we decided to provide the self-assessment tool.

Because most of the clinicians worked at university hospitals, most of them (11, or 84 percent) were engaged in teaching (lectures, instructional tutorials, and medical-assistant training). Most of them (10, or 76 percent) were interested in using interactive 3D graphics and providing 3D models, clinical imagery, and videos as complementary learning materials. So, we decided to provide the authoring tool.

The LiverAnatomyExplorer's Design

The LiverAnatomyExplorer pipeline comprises data acquisition, data preprocessing, data export, Web authoring, and Web-based visualization (see Figure 1).

Data Preprocessing

A crucial step in our pipeline is the generation and preparation of 2D and 3D visualizations. A typical medical dataset consists of a series of 2D images acquired using a technique such as CT (computed tomography) or MRI (magnetic resonance imaging). Typically, medical volume data are too large (hundreds of Mbytes) for online access and must be preprocessed and anonymized for online publishing.

Our dataset came from a database comprising several clinical cases with liver tumors, metastases, and vessel anomalies. (The database also contained surgical videos.) We used MeVisLab to import, convert, and prepare the cases. The preprocessing pipeline (see Figure 2) was straightforward using MeVisLab's image-processing and visualization techniques. The algorithms are encapsulated in connectable modules. Complex module networks can be wrapped in macro modules and enhanced by GUIs and Python scripting functions (for example, View2D).

First, we imported the 2D volume data (for example, MRI or CT data), using the `ImageLoad` module. We then used filtering modules to smooth the data.

Next, medical experts at MeVis Distant Services (www.mevis.de/loesungen_services_mds.html?&L=1) segmented the raw data's important structures. (For more on medical-image segmentation, see the related sidebar.) MeVisLab provides automatic segmentation algorithms (for example, thresholding and region growing) and semiautomatic techniques such as live-wire segmentation.[5] Live-wire segmentation is required if the liver or

Related Work in Web3D Technologies for Medical Education

HTML supports only the presentation of simple text, hyperlinks, and images. More flexible 2D vector graphics were added through SVG (Scalable Vector Graphics), an XML-based file format. Users can generate SVG graphics such as circles, rectangles, or polygons using drawing programs and edit them with any text editor. Frameworks such as Raphaël (http://raphaeljs.com) simplify embedding and using SVG objects in HTML. Raphaël is a common JavaScript toolkit for including SVG objects as DOM (Document Object Model) elements; it allows for cross-browser manipulation and animation of vector graphics.

The most widespread Web3D technology is VRML (Virtual Reality Modeling Language), which became an ISO (International Organization for Standardization) standard in 1997. It is still used but was replaced in 2004 by X3D (Extensible3D). X3D is a universal, device-independent ISO interchange format that represents 3D scenes and supports multiple data file encodings. To encode a 3D scene, you can use XML syntax, Open Inventor-like syntax derived from VRML97, or a compressed binary representation. X3D is organized into different profiles and components and supports multitextures and multimedia elements such as video and audio files. Of the other Web3D technologies, the best-known are Java 3D, Shockwave 3D, and QuickTime VR.

Since Web3D technologies' advent, researchers have deployed them to provide visualizations for medical education. Jianfeng Lu and his colleagues developed a VR learning environment based on VRML and VTK (Visualization Toolkit).[1] Reinhard Friedl and his colleagues used VRML, QuickTime VR, and MPEG videos in a visualization tool dealing with aortocoronary bypass grafting.[2] Angela Jerath and her colleagues created a Flash-based learning system that visualizes a polygonal heart model.[3] They constructed the model from transesophageal echocardiography images, using Osirix and Cinema 4D software. Esmitt Ramirez and Ernesto Coto described a general client-server architecture for Web-based 3D medical visualizations based on node.js and WebSockets.[4] A remote graphics server performs the 3D rendering. Jodi Crossingham and her colleagues developed a semi-interactive Flash-based tool for teaching liver anatomy based on a simulated healthy human liver.[5]

More recent Web applications, such as BioDigital Systems[6] and Zygote Body,[7] use WebGL (Web Graphics Library) to render interactive medical 3D scenes. (We describe WebGL in more detail later.) However, they employ simulated virtual models without anatomical variants or pathologies.

The Limitations of Current Web3D Medical Learning Systems

Existing Web3D medical learning systems have some important drawbacks. They often cover only a few cases and use 3D-modeling tools (for example, Maya, Cinema 4D, or 3ds Max) to reconstruct the anatomical situs. So, the 3D graphics don't represent realistic patient-specific anatomical variations or diseases.

Another important issue is the use of proprietary browser plug-ins—for example, Cosmo Player or BS Contact—to display Web-based VRML or X3D models. Especially in clinical environments, installing plug-ins is often restricted or even impossible for security reasons. Furthermore, plug-ins don't provide sufficient scalability and flexibility because they must take into account different platforms and operating systems.

Web3D techniques such as QuickTime VR or Java 3D don't provide the level of detail and interactivity that medical education requires. Server-based remote rendering also isn't promising because response times might be very long.[8] So, server-based renderings don't facilitate advanced real-time 3D visualizations. Moreover, Web3D medical learning systems rarely provide additional multimedia content and self-assessment tools along with the 3D models. However, these systems need to integrate such content and tools to enhance educational processes.[9] In addition, authoring tools that instructors can easily use to generate online student assessments of medical 2D and 3D visualizations aren't well established.

Another important limitation is nonexpert users' inability to navigate 3D visualizations. As Luca Chittaro and Roberto Ranon noted, "Users are often unable to move as they want, they get easily lost or do not know how to reach a particular location or point of view."[10] So, Web3D medical learning systems must provide easy-to-understand, simple-to-use 3D navigation controls and user interfaces.

HTML5, WebGL, and X3DOM

Prior Web3D medical learning systems' major drawbacks were security and incompatibility issues because users had to deal with unfamiliar browser plug-ins. HTML5 tries to overcome these issues through the `<canvas>` element. Using WebGL, a low-level JavaScript graphics API, hardware-accelerated 3D graphics can be rendered inside this element. WebGL is basically a JavaScript binding of OpenGL ES (Open Graphics Library for Embedded Systems) 2.0 and is implemented in such browsers as Firefox, Google Chrome, Opera, and Safari. WebGL is made by and designed for graphics experts who have an in-depth understanding of transformation matrices and texturing.

Most Web developers, however, don't have such a background and just want to share their 3D models. So, Johannes Behr and his colleagues proposed X3DOM, an integrated, abstract scene-graph layer that's directly mapped to DOM elements.[11] Using X3D's hierarchical scene-graph model as a visualization data structure has three main advantages:[11]

- X3D is already an ISO standard and defines XML encoding.
- You can integrate 3D scenes into DOM with just a few lines of code.

```
<html>
<head></head>
<body>
<x3d width=”600px” height=”600px”>
  <Scene>
    <Background skyColor=”0.8 0.8 0.8”></Background>
    <Viewpoint centerOfRotation=”0 0 0” position=”0 0 300.000”></Viewpoint>
    <NavigationInfo type=”EXAMINE”></NavigationInfo>
    <DirectionalLight ambientIntensity=”1.0” direction=”1 1 1”></DirectionalLight>
    <Transform id=”myScene” onclick=”alert(‘clicked’)”>
      <Shape>
        <Appearance>
          <Material diffuseColor=”1.0 0.8 0.5” transparency=”0.0”></Material>
        </Appearance>
        <IndexedTriangleSet colorPerVertex=”true” index=”0 1 2 3 2 1 4 ...”>
          <Coordinate point=”2.0 -41.1 -3.5 2.0 -40.6 ...”></Coordinate>
          <Normal vector=”-0.9 -0.2 -0.3 -0.9 ...”></Normal>
        </IndexedTriangleSet>
      </Shape>
    </Transform>
  </Scene>
</x3d>

<script type=”text/javascript”>
function changeColor()
{
  var object3D = $(“#myScene Material”);
  object3D.setAttribute(“diffuseColor”, “0 0 1.0”);
}
</script>
</body>
</html>
```

Figure A. An X3D scene integrated into HTML. The X3D nodes `<Background>`, `<Viewpoint>`, `<NavigationInfo>`, and `<DirectionalLight>` declare global scene settings. The scene consists of a `<Transform>` node and a `<Scene>` node followed by an `<Appearance>` node and an `<IndexedTriangleSet>` node to declare 3D parameters such as vertices, normals, colors, and transparencies.

- You can easily access and manipulate scene-graph elements, using JavaScript events and Cascading Style Sheets.

To export X3D-formatted 3D scenes, you can use well-known 3D-modeling tools such as Blender or medical-image-processing libraries such as MeVisLab.[12] Then, using X3DOM and WebGL, you can directly integrate the visualizations into HTML and render them in real time.

Figure A shows an X3D scene integrated into HTML. The X3D container can be scaled using width and height attributes. The X3D nodes `<Background>`, `<Viewpoint>`, `<NavigationInfo>`, and `<DirectionalLight>` declare global scene settings. The scene consists of a `<Transform>` node and a `<Scene>` node followed by an `<Appearance>` node and an `<IndexedTriangleSet>` node to declare 3D parameters such as vertices, normals, colors, and transparencies.

Using X3D and X3DOM lets you define JavaScript event functions for accessing particular X3D elements. For example, the function `changeColor()` updates the scene element's color on demand. You can also declare click events in the X3D code—for example, having the visualization react to a mouse click on a particular 3D object.

References

1. J. Lu et al., "Virtual Learning Environment for Medical Education Based on VRML and VTK," *Computers & Graphics*, vol. 29, no. 2, 2005, pp. 283–288.
2. R. Friedl et al., "Virtual Reality and 3D Visualizations in Heart Surgery Education," *Heart Surgery Forum*, vol. 5, no. 3, 2002, pp. 17–21.
3. A. Jerath et al., "An Interactive Online 3D Model of the Heart Assists in Learning Standard Transesophageal Echocardiography Views," *Canadian J. Anesthesiology*, vol. 58, no. 1, 2011, pp. 14–21.
4. E. Ramirez and E. Coto, "Web Visualization of 3D Medical Data with Open Source Software," *Proc. Conf. Academic Research and Industrial Meeting* (JIFI-EAI), Faculty of Eng., Universidad Central de Venezuela, 2012.
5. J.L. Crossingham et al., "Interpreting Three-Dimensional Structures from Two-Dimensional Images: A Web-Based Interactive 3D Teaching Model of Surgical Liver Anatomy," *HPB*, vol. 11, no. 6, 2009, pp. 523–528.
6. J. Qualter et al., "The Biodigital Human: A Web-Based 3D Platform for Medical Visualization and Education," *Medicine Meets Virtual Reality 19*, IOS Press, 2012, pp. 359–361.
7. R. Kelc, "Zygote Body: A New Interactive 3-Dimensional

Cont. on page 52

Related Work in Web3D Technologies for Medical Education (cont.)

Didactical Tool for Teaching Anatomy," *WebmedCentral Anatomy*, vol. 3, no. 1, 2012; www.webmedcentral.com/article_view/2889.
8. A.V. Poliakov et al., "Server-Based Approach to Web Visualization of Integrated Three-Dimensional Brain Imaging Data," *J. Am. Medical Informatics Assoc.*, vol. 12, no. 2, 2005, pp. 140–151.
9. N.W. John, "The Impact of Web3D Technologies on Medical Education and Training," *Computers & Education*, vol. 49, no. 1, 2007, pp. 43–51.
10. L. Chittaro and R. Ranon, "Web3D Technologies in Learning, Education and Training: Motivations, Issues, Opportunities," *Computers & Education*, vol. 49, no. 1, 2007, pp. 3–18.
11. J. Behr et al., "A Scalable Architecture for the HTML5/X3D Integration Model X3DOM," *Proc. 15th Int'l Conf. Web 3D Technology* (Web3D 10), ACM, 2010, pp. 185–194.
12. F. Ritter et al., "Medical Image Analysis: A Visual Approach," *IEEE Pulse*, vol. 6, no. 2, 2011, pp. 60–70.

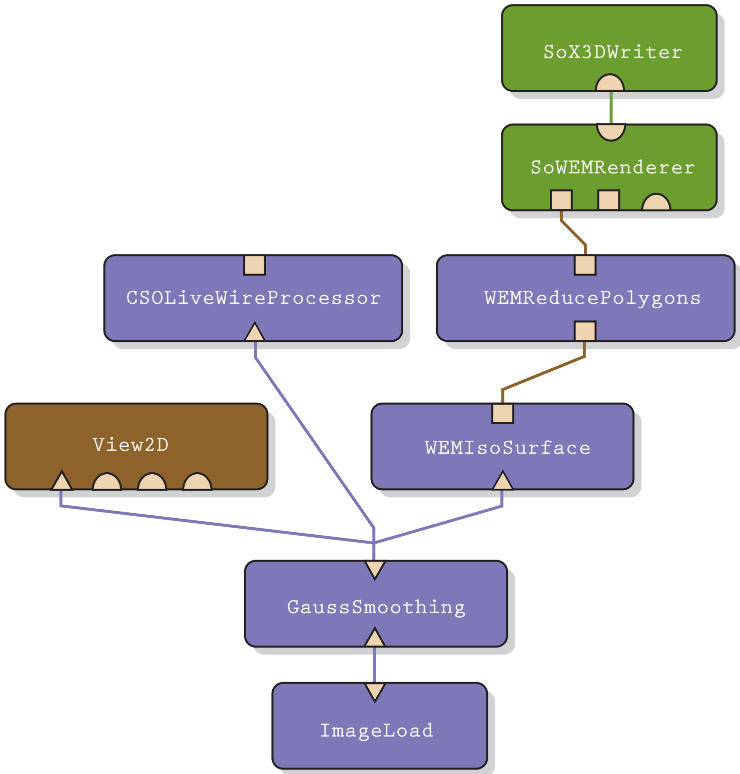


Figure 2. The MeVisLab preprocessing pipeline. Modules load, filter, and segment 2D medical volume data. The `SoX3DWriter` module exports the segmented areas to X3D.

its vessels are infiltrated by tumors, which would cause automatic methods to fail. Using live-wire segmentation and their expert knowledge, the medical experts segmented the liver, veins, arteries, and tumors.

We then converted the 2D segmentation masks to 3D surface meshes using the `WEMIsoSurface` module and the marching-cubes algorithm. The surface models' size is more crucial for Web-based rendering than for local use. So, automatic mesh simplification algorithms based on quadric error matrices reduced the number of polygons. We set the reduction to 80 percent to ensure small file sizes without losing relevant information. The simplified surface mesh took up approximately 5 to 10 Mbytes (approximately 100,000 polygons, 20 to 25 fps, tested on a 2.3-GHz Intel Core i5 with 4 Gbytes of RAM and an Nvidia Geforce GT 540M). This enabled real-time Web-based rendering in clinical environments.

Exporting Data

We exported the segmented structures in a Web-compatible format. To achieve fast online access of anonymized patient data, we chose JPEG as the format for 2D images. Using compressed JPEG images instead of highly detailed raw data is reasonable because no exact diagnostics have to be accomplished on the Web platform. The segmented areas were automatically exported as SVG polygons and wrapped in an XML file, using MeVisLab.

We exported the 3D surface mesh as an X3D file.

The Authoring Tool

This wizard-based tool lets instructors upload, edit, and delete cases. Uploading a case involves six steps. First, the instructor enters the anonymized patient information (age, gender, diagnosis, and surgical indication) and the case's learning goals.

Second, the authoring tool prompts the instructor to upload all image data (JPEG) and segmentations (SVG) provided by MeVisLab during preprocessing (see the section "Data Preprocessing").

Third, the instructor uploads the generated X3D file. The LiverAnatomyExplorer automatically converts X3D models into the JSON format to speed up loading times and save Web space. JSON is a lightweight, text-based interchange format useful for serializing and transmitting structured data over network connections. Unlike XML, JSON-formatted data don't need to be parsed to access particular scene nodes (viewpoints, lighting materials, vertices, and so on). The scene structures can be used directly as JavaScript objects in Web environments.

Fourth, the instructor creates the quiz (see Figure 3). First, the instructor chooses a question type and enters a question. There are three types of questions: multiple choice, 2D click, and 3D click. Next, he or she enters the answer options.

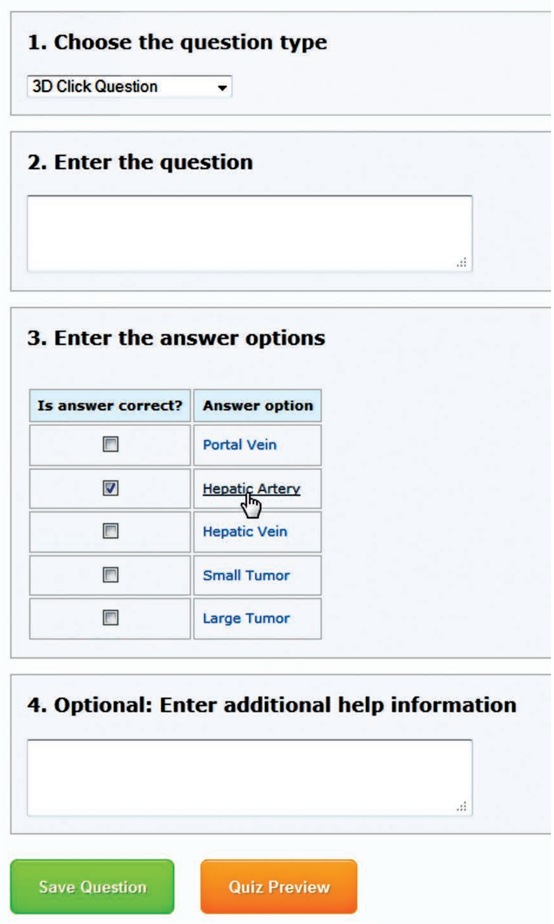


Figure 3. The dialog box for creating a quiz question. There are three types of questions: multiple choice, 2D click, and 3D click.

For example, if the instructor chooses a 3D click question, the tool names each structure of the 3D model, links that structure to the corresponding structure in the 3D viewer, and represents it with a check box. The instructor can also enter text information for each question—for example, hyperlinks to interesting websites. After saving the question in a database, the instructor can enter a new one.

In the fifth step, the instructor can add multimedia files such as videos showing cancer therapy options or the use of novel surgical instruments.

Finally, the instructor previews the case if necessary and finishes it. Before finishing it, the instructor enters a name and the difficulty level. After the instructor saves the case in the database, students can access it.

What Is Medical-Image Segmentation?

Each slice of a medical dataset represents a portion of the body at regular intervals. These raw images must be processed to identify anatomical structures—for example, tumors or vascular structures—and exclude irrelevant structures such as bones. Such segmentation can be performed either manually with drawing tools or semiautomatically with algorithms using model assumptions regarding shapes and gray values. Many image segmentation and visualization tools—for example, MeVisLab, VTK (Visualization Toolkit), and Amira—provide such algorithms. Segmentation usually produces a binary mask of an anatomical structure that can serve as an overlay on the original data.

The Learning Module

This module presents several realistic patient cases with rising difficulty levels. Each case comprises five steps that students can navigate individually (see Figure 4).

Learning content and targets. The LiverAnatomyExplorer first presents the learning content and objectives. There are three main objectives: the student will

- internalize that the liver has a highly variable shape, especially if it's affected by tumors;
- identify and estimate abnormal liver vessels, their territories, and resection strategies; and
- learn how to explore each virtual 3D model.

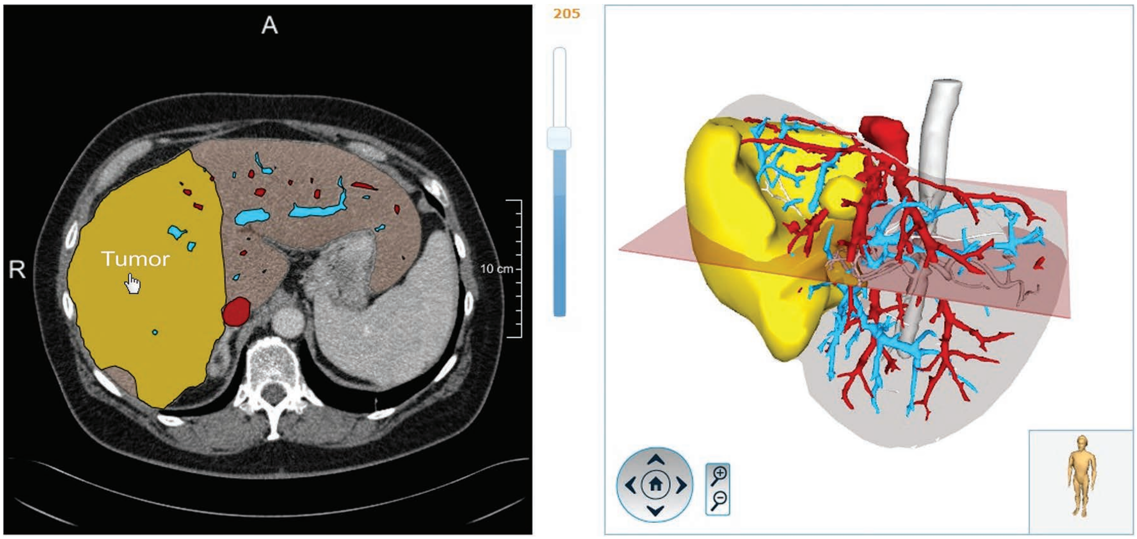
We verified these objectives with our clinical partners.

Patient information. Next, the LiverAnatomyExplorer provides clinical background: the anonymized patient data, information about radiological findings, and the patient's previous diagnoses and therapies.

The 2D viewer. We developed a plug-in-free 2D image viewer based on HTML, SVG, and JavaScript. The exported JPEG images of the medical volume dataset are loaded on demand; students can process them slice-by-slice in real time, using a slider.

The 2D viewer loads the segmented structures and maps them as colored overlays on the JPEG images, using the Raphaël SVG framework. (For more on Raphaël, see the sidebar "Related Work in Web3D

Figure 4. The steps of the LiverAnatomyExplorer learning module, which presents several realistic patient cases with rising difficulty levels.



(a) (b)

Figure 5. Synchronized (a) 2D and (b) 3D viewers. Changing the slider position updates both the *z* position of the 2D CT slice stack and the rectangular plane in the 3D viewer. So, students don't lose the orientation in the 3D scene. In the 3D viewer, students can employ a 3D widget (on the bottom left) to easily rotate and scale the 3D model. An abstract orientation model (on the bottom right) illustrates the overall viewpoint on the scene.

Technologies for Medical Education.") Using SVG objects allows for flexible access, scaling, and coloring of the overlays. Mouseover events on the SVG objects display text annotations (see Figure 5a). By using Raphaël, we can employ JavaScript `onclick` events to enhance the overlay objects. If the student clicks on a certain segmented structure, the 2D viewer can present additional information or hyperlinks concerning that object. Students can enable or disable the colored vector graphics through check boxes next to the viewer. Moreover, silhouettes enhance the segmented objects' boundaries.

The 3D viewer. The LiverAnatomyExplorer's most important feature is the 3D viewer, which is based on X3D, WebGL, and an X3DOM JavaScript framework. To ensure fast access to the overall website, the 3D viewer doesn't embed 3D models directly in HTML. Instead, before it includes the 3D content, PHP scripts automatically convert the X3D file into a JSON (JavaScript Object Notification) file on the webserver, as we mentioned before.

The 3D viewer dynamically accesses the JSON file's 3D scene nodes using AJAX (Asynchronous JavaScript and XML) events on demand. The viewer downloads the nodes from the Web server and directly integrates them into the client website. This process takes just a few seconds, depending on the Internet connection.

Students can freely rotate, translate, and zoom in and out on the WebGL-rendered 3D scene without installing a plug-in. The 3D viewer renders the liver surface mesh semitransparently to enable evaluation of arteries and veins. Different opaque colors distinguish the vessel's polygonal meshes. Because the X3D runtime is seamlessly integrated into the DOM (Document Object Model) using X3DOM, scene nodes can be accessed and changed in real time. For example, individual 3D objects can be set as transparent or opaque by manipulating the X3D `<Material>` node's transparency attribute. Moreover, the DOM integration enables styling the X3D content using Cascading Style Sheets. For example, the 3D scene can be scaled using width and height attributes, or the scene can be placed at a desired position on the website.

One major advantage of using SVG and WebGL is the ability to combine both methods. Clinical experts are familiar with slice-based 2D visualizations of anatomical structures. In contrast, medical students are often unfamiliar with interactive 3D graphics and might get lost in complex 3D scenes. So, we place the 2D and 3D viewers side by side (see Figure 5). The slice's position in the 2D viewer is synchronously represented by a semitransparent red plane in the 3D model. The LiverAnatomyExplorer achieves live synchronization by updating the 3D model's world matrix according to the slider's position.

Enhanced user interaction. To smoothly rotate the complex 3D scene, students employ a 3D widget on the 3D viewer's lower left (see Figure 5b). It was inspired by the well-known Google Maps user interface metaphor of a circular touch panel with clickable buttons. With this widget, students can rotate the 3D scene vertically or horizontally in fixed 30° intervals around the *x*- or *y*-axis.

To zoom in or out, students can either click the right mouse button and drag the mouse vertically or click on the widget's lens buttons. Clicking on the home icon resets the initial viewpoint. Furthermore, the 3D viewer presents a greatly simplified orientation model of a human body to illustrate the overall viewpoint on the scene (see Figure 5b). This polygonal model is embedded as a second X3D scene, and its camera parameters update synchronously as the main scene's camera position changes.

An essential objective of medical learning systems is for students to correctly identify anatomical structures. So, we implemented a tooltip function (see Figure 6). When students place the cursor over a 3D structure, the correct medical term for that structure appears. In addition, the 3D object's `emissiveColor` increases slightly to highlight the mouseover effect (`emissiveColor` is an X3D component).

Quizzes, feedback, and multimedia. Immediate customized feedback is essential for students to gain knowledge and to monitor their progress.[1] So, a multifunctional self-assessment tool appears next to the 2D and 3D viewers; students can use it to test their liver anatomy knowledge.

The self-assessment tool presents multiple-choice, 2D-click, and 3D-click questions, as we mentioned before. A typical question might be, "Please identify the portal vein in the liver." To fulfill this task, students must click on the correct structure. One way we achieve this is by setting JavaScript `onclick` events, on both 2D SVG elements and 3D objects.

For multiple-choice questions, the learning module presents text indicating whether the students' answers are correct or incorrect. For click questions, it presents a blinking animation of the correct answer in the 2D or 3D scene. If students answer at least 75 percent of the questions correctly, they can investigate a new, more difficult case.

The learning module provides hyperlinks to relevant websites. Furthermore, students can view a high-quality surgical video composed and annotated by an expert surgeon. So, they can gain insight into surgical problems such as potential complications or special resection techniques that had to be used for this case.

Surgical training. Besides teaching anatomical structures, the learning module can train students on surgical procedures. In daily clinical routine, a liver surgeon must resect tumor-affected regions with respect to surrounding blood vessels and organ boundaries. Computer-generated 3D visualizations can support such treatment decisions.

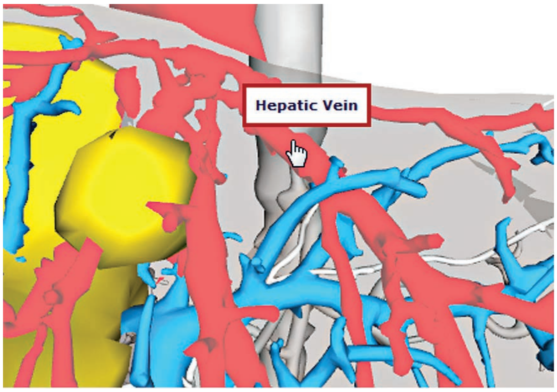


Figure 6. When students place the cursor over a 3D structure, the correct medical term for that structure appears.

So, we integrated several *resection proposals* that medical experts derived from individual patient anatomy. A resection proposal consists of a resection volume and a remnant volume, visually separated by a resection plane (see the bottom-right image in Figure 7). The remnant is the part of the liver that must be saved. It should be as large as possible and tumor-free to assure a curative patient treatment. The student interactively explores the 3D model and decides which proposed treatment option is indicated. After choosing a resection method, the student gets immediate feedback on his or her answer.

Evaluation

We performed two evaluations of the LiverAnatomyExplorer.

The Informal Evaluation

A radiologist and a surgeon evaluated the LiverAnatomyExplorer. The radiologist indicated that the learning module's most promising feature was the free exploration of high-quality 3D models derived from individual patient data. She adroitly explored the 2D and 3D visualizations.

The surgeon favored the 3D widget for interacting with the 3D model and resetting distorted viewpoints. He recommended integrating additional anatomical structures, such as bile ducts, into the visualization to provide a better understanding of the anatomical location. He also suggested that we make the learning content more granular and separate it for two target groups: medical students and assistant doctors.

The Formal Evaluation

According to the International Organization for Standardization (ISO), the user experience includes not only the users' emotions, satisfaction, and fun

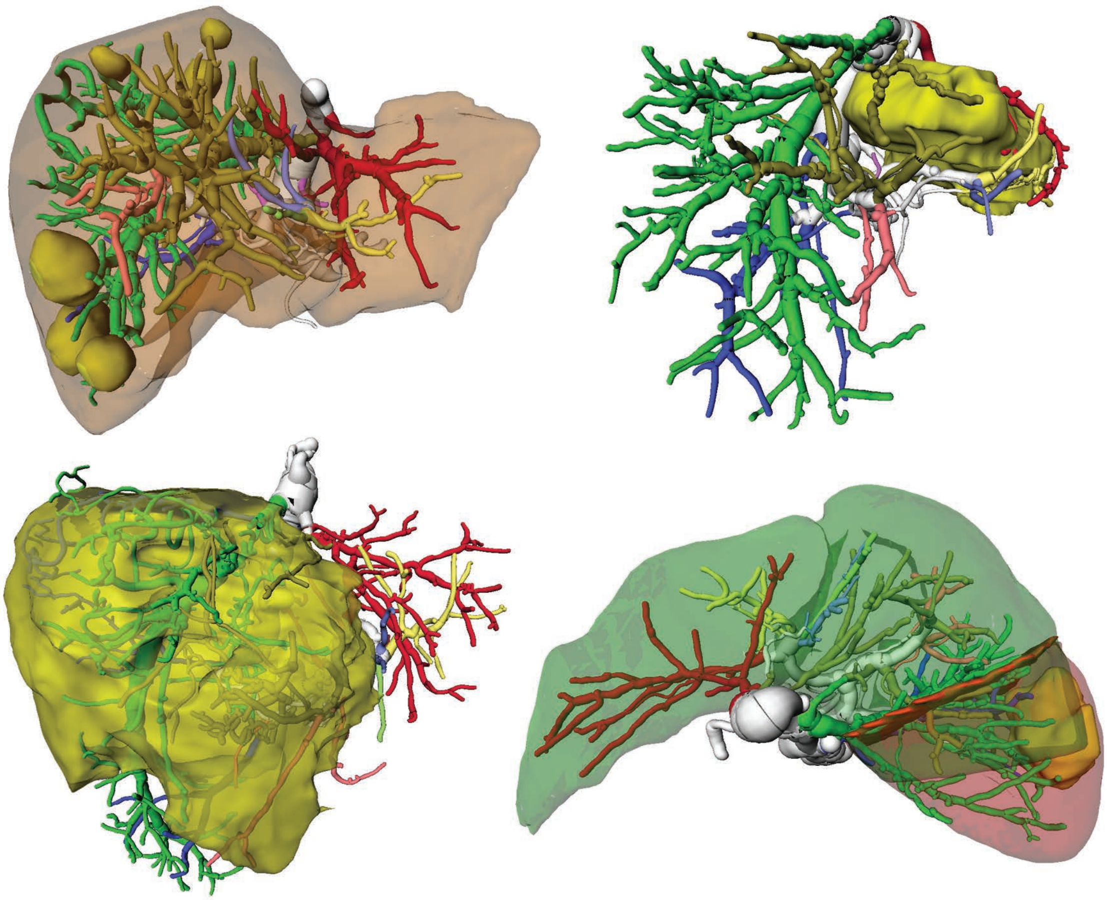


Figure 7. Examples of visualized patient datasets. We've integrated different patient-specific liver datasets with varying anatomy into the LiverAnatomyExplorer. Patients had variable tumor sizes and metastases. The bottom-right image is a 3D resection proposal displayed in our WebGL-based 3D viewer. The remnant (the part of the liver that must be saved) is green, the resection plane is orange, and the resection volume is red.

but also aesthetic and usability aspects.[6] To measure these soft values and gain insight into users' beliefs, we had medical students participate in a study.

The procedure. The study involved 40 female and 14 male students. Their average age was 24, and they were in the seventh to ninth semester of their training. After a short introduction (approximately 5 minutes) to the LiverAnatomyExplorer, the participants took a short instructional tutorial and performed the first case. Then, they filled out an online survey assessing their user experience and the tool's learning aspects (see Figure 8). They responded to each survey statement using a five-point Likert scale (1 = strongly disagree and 5 = strongly agree). They could also add comments.

The results. The survey results (see Figure 9) and the participants' verbal comments indicate that the LiverAnatomyExplorer provided an overall good user experience. According to the participants,

- the website was attractive (S1's average score was 3.94, with a standard deviation [σ] of ±0.62),
- the navigation was self-explanatory (S2's average score was 4.22, with σ = ±0.63), and
- waiting times were low (S3's average score was 4.38, with σ = ±0.81).

The participants had few to no problems handling the interactive 2D slice data and 3D models. This might be because of the instructional tutorial and the orientation guide, which many participants found helpful. (One participant said, "The small man icon is very useful because it provides me an overview of the 3D scene. The home icon is useful for restoring the 3D viewpoint.") However, some participants stated problems with precisely selecting particular 3D structures, mostly very thin liver vessels. Zooming in on the scene resolved this issue.

The participants rated the learning aspects as good, with some exceptions. They highly rated

- the content's reality (S10's average score was 3.98, with σ = ±0.68) and
- the knowledge they gained because of the multimodal data (S11's average score was 3.96, with σ = ±0.91).

In contrast, some participants exposed the learning module's limitations. One participant would have liked to "use the tool for other organs, not only the liver." Another participant mentioned that the "gall bladder is not visible in the datasets."

Interestingly, answering the anatomy questions was a big challenge, resulting in an average score of 2.85 (σ = ±0.56) for S12. This is because the questions concerning liver segments and vessel anomalies turned out to be too specialized for medical students at this education level. One participant said, "This could be perfect for assistant doctors; for me, it's too difficult!"

Most participants liked the immediate feedback (S13), which received an average score of 3.50 (σ = ±0.77). However, some participants complained about the feedback's low level of detail. The overall learning success (S14) received an average score of 3.59 (σ = ±0.69). Many participants commented on how the tool benefited their learning progress. For example, one participant said, "If you have shortcomings in a special domain, you can repeat the exercise." Another participant stated, "It is nice to have a noncommercial e-learning tool for this special domain. This tool enables more interactivity and knowledge gain compared to typical textbooks!"

Usability and User Experience Aspects

S1. The website has an attractive design.
S2. The navigation is comprehensible.
S3. The website has a fast response time.
S4. The 2D slice data can be easily handled (slicing and enabling or disabling structures).
S5. The 3D models can be easily handled (rotating, zooming, and pointing).
S6. I find the 3D orientation guide and the 3D widget useful.
S7. I find the instructional-tutorial mode useful.
S8. I have fun using the LiverAnatomyExplorer.

Learning Aspects

S9. The learning contents fulfill my needs.
S10. The learning contents are realistic.
S11. The multimodal data help me to get a better understanding of individual liver anatomy.
S12. Answering the learning questions is easy.
S13. The learning feedback is appropriate.
S14. The subjective learning success is high. [That is, the participant felt that he or she had easily learned the presented information.]

Figure 8. User study survey statements. The participants responded to each statement using a five-point Likert scale (1 = strongly disagree and 5 = strongly agree). They could also add comments.

The LiverAnatomyExplorer isn't just for teaching liver anatomy. Because we use the X3D format for exporting 3D models, the architecture is easily adaptable to other organs. Moreover, the architecture enables adding use cases. Installing additional software in operating rooms is usually prohibited for security reasons. So, the plug-in-free WebGL technology provides promising advantages for displaying 3D visualizations during surgery. WebGL might enable surgeons to compare the current anatomical situs with a virtual 3D view of the intended resection strategy.

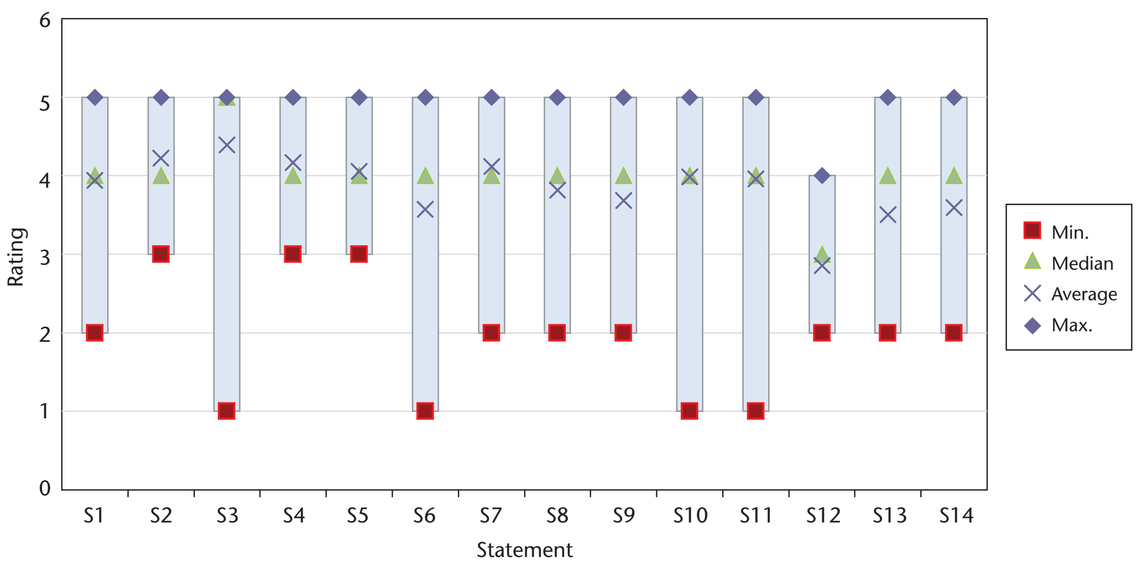


Figure 9. The results of a study of 54 medical students concerning the LiverAnatomyExplorer's user experience and learning aspects, for the survey statements in Figure 8. A response of 1 means "strongly disagree"; a response of 5 means "strongly agree." The results indicate that the LiverAnatomyExplorer provided an overall good user experience.

Furthermore, WebGL enables 3D renderings on portable devices such as smartphones and tablets. Doctors could use this capability to enhance patient interviews during ward rounds. High-quality surface meshes, videos, and animations could help provide comprehensible explanations of planned surgery.

Surgical assistants greatly benefit from operation simulation software. It would be desirable to provide training systems such as the LiverSurgeryTrainer[7] online using WebGL. For example, such an approach would let assistants define and modify resection planes and evaluate operation strategies validated by experts, using a common Web browser.[8] This could be enhanced by 3D input devices with haptic force feedback to realistically simulate surgery conditions.[9]

Medical e-learning enables ubiquitous, just-in-time individualized continuing education. However, to avoid students' social isolation, online discussion groups and feedback mechanisms are necessary to foster student-instructor communication.[1] We're working on a Web 2.0 surgical-collaboration platform. It aims to enable access to user-generated teaching materials, share medical knowledge, and promote collaborative discussion of novel surgical techniques and equipment among experts, assistants, and students.

Acknowledgments

We thank Karl-Jürgen Oldhafer (Asklepios Clinic Barmbek, Hamburg) for his clinical support and the Fraunhofer Institute for Medical Image Computing for advanced MeVisLab features. The German Federal Ministry of Industry and Research supported this research in the framework of the SurgeryTube project (promotional reference 01PF08003E).

References

1. R. Ellaway and K. Masters, "AMEE Guide 32: E-learning in Medical Education Part 1: Learning, Teaching and Assessment," *Medical Teacher*, vol. 30, no. 5, 2008, pp. 455–473.
2. D.A. Cook, "Web-Based Learning: Pros, Cons and Controversies," *Clinical Medicine*, vol. 7, no. 1, 2007, pp. 37–42.
3. S. Birr et al., "A Novel Real-Time Web3D Surgical Teaching Tool Based on WebGL," *Bildverarbeitung für die Medizin 2012*, Springer, 2012, pp. 404–409.
4. C. Pape-Köhler et al., "E-learning in Surgical Procedure Manuals and Blogs," *Der Chirurg*, vol. 81, no. 1, 2010, pp. 14–18 (in German).
5. E.N. Mortensen and W.A. Barrett, "Interactive Segmentation with Intelligent Scissors," *Graphical Models and Image Processing*, vol. 60, no. 5, 1998, pp. 349–384.
6. *Ergonomics of Human System Interaction—Part 210: Human-Centred Design for Interactive Systems*, Int'l Org. for Standardization, 2010.
7. J. Mönch et al., "The LiverSurgeryTrainer: Training of Computer-Based Planning in Liver Resection Surgery," to be published in *Int'l J. Computer Assisted Radiology and Surgery*, 2013.
8. G. Mani and W. Li, "3D Web Based Surgical Training through Comparative Analysis," to be published in *Proc. 18th ACM Int'l Conf. 3D Web Technology*, ACM, 2013.
9. Y. Jung et al., "Using X3D for Medical Training Simulations," *Proc. 13th Int'l Symp. 3D Web Technology* (Web3D 08), ACM, 2008, pp. 43–51.

***Steven Birr** is a research associate in the University of Magdeburg's Department of Simulation and Graphics. His research activities span Web-based e-learning applications, surgical education, and human-computer interaction. Birr received a Diploma in computer science from the University of Magdeburg. He's a member of the German Usability Professionals' Association. Contact him at steven.birr@ovgu.de.*

***Jeanette Mönch** is a research associate in the University of Magdeburg's Department of Simulation and Graphics. Her research interests are human-computer interaction and medical education. Mönch received a doctorate in computer science from the University of Magdeburg. Contact her at jeanette.moench@ovgu.de.*

***Dirk Sommerfeld** is a student research assistant in the University of Magdeburg's Department of Simulation and Graphics. His research interests are human-computer interaction and medical education. Sommerfeld received a Diploma in computer science from the University of Magdeburg. Contact him at desomku@googlemail.com.*

***Uta Preim** is a board-certified radiologist in Clinical Center Magdeburg's Department of Radiology. Her research interests are cardiac imaging, cardiac blood flow, and perfusion imaging. Preim received an MD from the University of Magdeburg. Contact her at uta.preim@med.ovgu.de.*

***Bernhard Preim** is a professor of computer science and the head of the visualization group in the University of Magdeburg's Department of Simulation and Graphics. His research interests are visualization techniques for medical volume data, 3D interaction techniques, and computer support for medical diagnosis and therapy planning. Preim received a habilitation in computer science from the University of Bremen. Contact him at bernhard.preim@ovgu.de.*