

A Practical Workflow for Making Anatomical Atlases for Biological Research

Yong Wan, A. Kelsey Lewis, Mary Colasanto, Mark van Langeveld, Gabrielle Kardon, and Charles Hansen ■ *University of Utah*

An anatomical atlas provides a detailed map for medical and biological studies of anatomy. These atlases are important for understanding normal anatomy and the development and function of structures, and for determining the etiology of congenital abnormalities. Unfortunately, for biologists, generating such atlases

is difficult, especially ones with the informative content and aesthetic quality that characterize human anatomy atlases. Building such atlases requires knowledge of the species being studied and experience with an art form that can faithfully record and present this knowledge, both of which require extensive training in considerably different fields. (For some background on anatomical atlases, see the related sidebar.)

With the latest innovations in data acquisition and computing techniques, atlas building

has changed dramatically. We can now create atlases from 3D images of biological specimens, allowing for high-quality, faithful representations. Labeling of structures using fluorescently tagged antibodies, confocal 3D scanning of these labeled structures, volume rendering, segmentation, and surface reconstruction techniques all promise solutions to the problem of building atlases.

However, biology researchers still ask, “Is there a set of tools we can use or a practical workflow we can follow so that we can easily build models from our biological data?” To help answer this question, computer scientists have developed many algorithms, tools, and program codes. Unfortunately, most of these researchers have tackled only one aspect of the problem or provided solutions to special cases. So, the general question of how to build anatomical atlases remains unanswered.

For a satisfactory answer, biologists need a practical workflow they can easily adapt for different applications. In addition, reliable tools that can fit into the workflow must be readily available. Finally, examples using the workflow and tools to build anatomical atlases would demonstrate these resources’ utility for biological research.

To build a mouse limb atlas for studying the development of the limb musculoskeletal system, University of Utah biologists, artists, and computer scientists have designed a generalized workflow for generating anatomical atlases. We adapted it from a CG artist’s workflow of building 3D models for animated films and video games. The tools we used to build the atlas were mostly commercial, industry-standard software packages. Having been developed, tested, and employed for industrial use for decades, CG artists’ workflow and tools, with certain adaptations, are the most suitable for making high-quality anatomical atlases, especially under strict budgetary and time limits. Biological researchers have been largely unaware

The anatomical atlas has been at the intersection of science and art for centuries. With certain adaptations, the computer graphics artist’s workflow and tools are practical for building high-quality atlases, essential to biological and medical research. Using this workflow, researchers built an atlas of the mouse limb musculoskeletal system.

A Brief Introduction to Anatomical Atlases

Truly an arena where science meets art, anatomical atlases evolved as technologies advanced in both fields: painting, printing, photography, microscopy, tomography, and computer graphics. Whenever a novel technology emerges, our knowledge of anatomy is enriched with both exciting scientific findings and the increasingly detailed information presented in atlases.

Historically, an anatomical atlas has been a book of illustrations and text that systematically explains a particular species' anatomy. Naturally, human anatomy has been the most studied. Atlases of human anatomy date back to such renowned anatomists as Vesalius, Leonardo da Vinci, William Hunter, and Henry Gray. Their creations are esteemed not only for their scientific value but also as masterpieces of art. The most influential printed human anatomy atlases available are Gray's,¹ Frank Netter's,² and Thieme's.³

The production and appearance of printed atlases have changed with the development of technology. Henry Vandyke Carter, the illustrator of Gray's book, drafted his illustrations in reverse on boxwood, which was then engraved for printing.⁴ Netter enjoyed the convenience brought by photography and modern printing. He chose gouache, a painting technique that could better render the highlights and shadows to give his illustrations a more 3D look. The illustrations of Thieme's *Atlas of Anatomy* were mostly hand drawn with Adobe Photoshop by Markus Voll and Karl Wesker. Although they largely follow the styles established by their predecessors, the use of digital media gives their illustrations finer details, smoother tonal gradations, and better transparency effects. All these contribute to the clear representation of particular anatomical features.

Printed atlases of other biological species are relatively scarce, and those in existence are usually decades old. For example, Eunice Greene's *Anatomy of the Rat*⁵ was published in 1935 and is still considered the definitive text for identifying rodent anatomical structures. In particular, Greene's atlas serves as the anatomy text for the laboratory mouse, one of the most important model organisms in biology and medicine.

Anatomical atlases are crucial to understanding normal anatomy and identifying congenital abnormalities. For educational purposes, physical models are also sometimes built and used. They provide a unique 3D model of anatomy but

are difficult to make, store, and maintain. They hardly ever achieve the level of detail required by scientific research and are rarely used in biological and medical research.

Computer graphics transformed anatomical atlases. Computer-generated atlases allow for a 3D, manipulable visualization of anatomy. Several 3D atlases have been generated for human anatomy, including the Visible Body (www.visiblebody.com), Cyber-Anatomy (www.cyber-anatomy.com), and Zygote's anatomical-model library (www.zygote.com). Because 3D models are commonly built by referencing 2D illustrations of anatomy books, building 3D atlases is expensive and requires specialized personnel with experience in digital modeling and anatomy.

For biologists to build 3D anatomical models, using 3D scanned biological samples is the most practical approach. Owing to noise and limited resolution, anatomical atlases that directly use volume renderings of scanned 3D volumetric data⁶ usually can't achieve the clarity of polygon-based atlases, composed of objects modeled by representing their surfaces with polygons. There are several published polygon-based anatomical atlases for biological research (for example, by Tao Ju⁷ and April DeLaurier and her colleagues⁸). However, easy-to-follow workflows and examples are unavailable for biologists to learn to make such atlases.

References

1. H. Gray and H.V. Carter, *Anatomy: Descriptive and Surgical*, 1st ed., J.W. Parker, 1858.
2. F.H. Netter, *Atlas of Human Anatomy*, 5th ed., Elsevier, 2011.
3. A.M. Gilroy, B.R. MacPherson, and L.M. Ross, *Atlas of Anatomy*, 1st ed., Thieme Medical Publishers, 2009.
4. R. Richardson, *The Making of Mr. Gray's Anatomy*, 1st ed., Oxford Univ. Press, 2008.
5. E.C. Greene, *Anatomy of the Rat*, reprint, Hafner, 1968.
6. K.H. Höhne et al., "A Volume-Based Anatomical Atlas," *IEEE Computer Graphics & Applications*, vol. 12, no. 4, 1992, pp. 72–78.
7. T. Ju, "Building a 3D Atlas of the Mouse Brain," PhD thesis, Dept. of Computer Science, Rice Univ., 2005.
8. A. DeLaurier et al., "The Mouse Limb Anatomy Atlas: An Interactive 3D Tool for Studying Embryonic Limb Patterning," *BMC Developmental Biology*, vol. 8, 2008, article 83.

of these resources. By describing our experiences in this project, we hope to show biologists how to use these resources to make anatomically accurate, high-quality, and useful anatomical atlases.

Data Acquisition

The biologists who worked on the mouse limb atlas are researching the cellular and molecular mechanisms governing the patterning and assembly of

the musculoskeletal system during development. Understanding how the musculoskeletal system is assembled is a fundamental question in developmental biology. In addition, congenital defects in limb and musculoskeletal development are relatively common in humans; understanding these defects' etiology is of interest to the medical community.

Mice are the primary model organism used to study limb musculoskeletal development. Not only

Confocal Laser Scanning Microscopy

To capture the finely detailed structures from a biological sample, such as a mouse limb, researchers tag structures using different fluorescently labeled antibodies that bind only to particular structures (for example, tendons, muscles, and nerves). A confocal microscope excites the fluorescent tags with lasers, and different detectors collect the emitted light from the tags. So, one detector will collect the image of tendons (in green), another will collect muscles (in red), and another will collect nerves (in blue).

The confocal microscope collects a series of images (each an optical thin section) by progressively moving the xy focal plane and sequentially sampling through the limb's z depth. Thus, it generates a stack of in-register 2D images, and each 2D xy image contains three color channels (green with tendons, red with muscle, and blue with nerves). Scanning can be time-consuming for large samples, so the z increment between optical sections is often increased. Thus, the images produced often have high resolution in the xy -plane but lower resolution in the xz - and yz -planes (see Figure 1 in the main article). Although this technique can capture details well, confocal data might require digital processing before visualization or analysis, owing to a low signal-to-noise ratio.

are mouse and human development similar, but also many genetic tools (such as the ability to create *knockout mice*—mice with a gene inactivated) and molecular reagents are available with mice.

To facilitate studying mouse limb development, the biologists wanted to create a 3D atlas that clearly displayed bones, tendons, muscles, and nerves. April DeLaurier and her colleagues had previously published a mouse limb atlas.¹ However, that atlas displayed just muscles and bones and lacked details of the muscle fibers' orientation.

To construct the limb atlas, we obtained digital images of the musculoskeletal system of mouse limbs. First, in each limb, we labeled tendons, muscles, and nerves with different fluorescently tagged antibodies. We then used *confocal laser scanning microscopy* (CLSM; see the related sidebar) to image the tagged limbs. For each limb, we obtained a stack of in-register optical thin sections showing tendons in green, muscles in red, and nerves in blue. Bones were distinct black regions in the green and red channels.

Figure 1 shows an acquired dataset of a mouse hind limb, which we visualized by volume rendering with FluoRender (see the “Our Tools” sidebar).

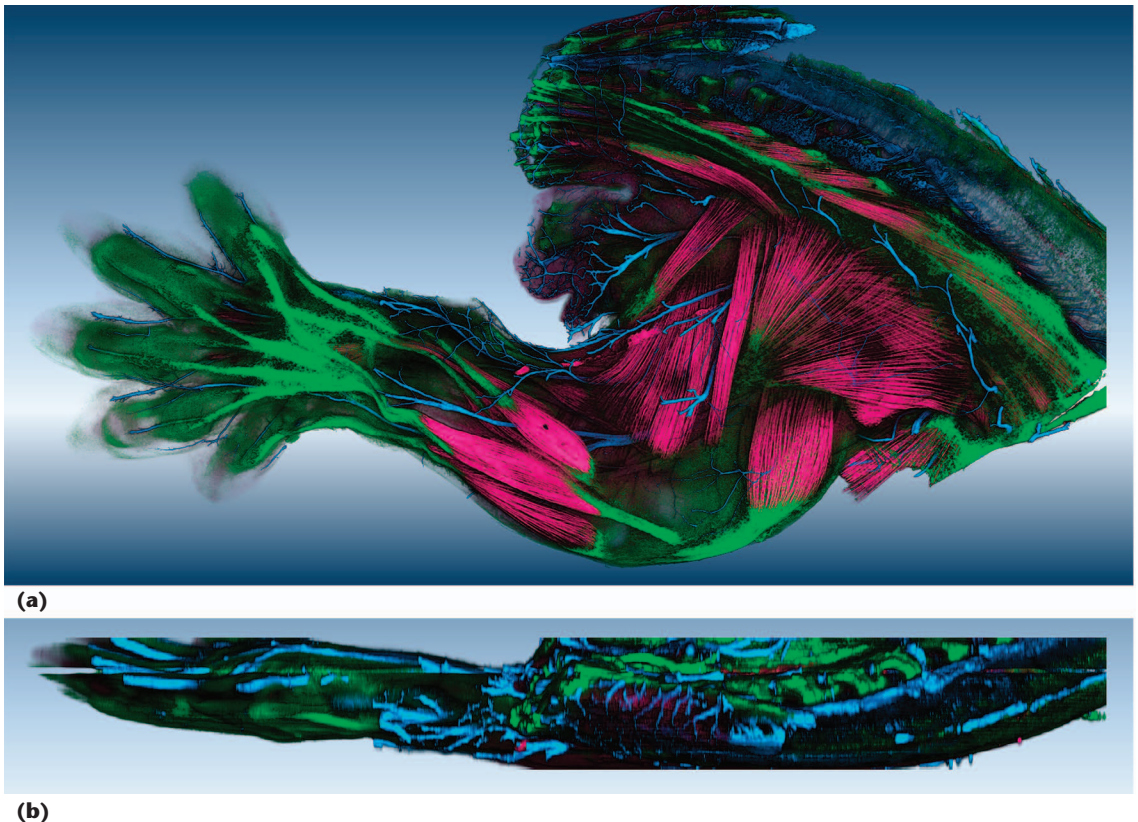


Figure 1. A volume-rendered visualization (using FluoRender) of the confocal data acquired for our atlas-building project. This is the hind limb of a 14.5-day-old embryonic mouse. Muscles are red, tendons green, and nerves blue. (a) In the xy -plane view, the visualization contains rich details of the structures. (b) In the xz -plane view, the visualization becomes coarse owing to the increased z increment.

Our Tools

Our workflow for creating anatomical atlases employed the following tools.

Photoshop

Since its first appearance in 1988, Photoshop (www.photoshop.com) has attained an unquestionable position in image editing and processing. Many biologists also use it for research. We mainly used layers, the levels tool, and the quick-selection tool.

The quick-selection tool speeds up selection of local features. Although the tool's exact algorithm is unpublished, we infer from its behavior that it most likely calculates the image's gradient field and uses region-growing or front-propagation algorithms to progressively detect boundaries. Jiangyu Liu and colleagues described the algorithm of a similar implementation.¹ Photoshop also provides a collection of brushes, which we used to generate the stencil patterns for texturing.

FluoRender

FluoRender (www.fluorender.com) is a software package we developed at the University of Utah for visualizing confocal-microscopy data. Recently we added selection tools that can easily extract neural structures from confocal data. The design of FluoRender's selection tool is similar to that of Photoshop's quick-selection tool. However, we used it to isolate structures from volume-rendered data instead of 2D slices, which made it suitable for extracting complex 3D structures. FluoRender also supports rendering semitransparent polygon models, which made it a good candidate for presenting the atlas data.

Maya

Bestowed with the technologies from the Advanced Visualizer, PowerAnimator, and StudioTools, Maya (usa.autodesk.com/maya) is routinely used in the film industry. We used Maya's polygon-modeling tools to build models and fit

them to extracted structures' shapes. Its flexibility also enabled us to write a script for visualizing volumetric data within its user interface.

Mudbox

For both digital sculpting and painting, Mudbox (autodesk.com/mudbox) has a simple, intuitive user interface for interacting with 3D polygon models. We used its painting functions mainly to generate textures for polygon models.

Comments

For certain features that commercial tools didn't offer, such as rendering and segmentation of volumetric data, FluoRender was the only academic tool we used. Compared to many specialized tools used in biological data processing, our chosen tools were stable, intuitive, and easy to acquire. Photoshop was the only tool that required purchasing; it costs US\$100 to \$200 for academic use. The other tools were free or had free academic licenses. The popularity of the artists' tools also makes it easy to find online resources and tutorials as well as skilled users. Data exchange among these tools is easy and supported by commonly used file formats. So, we could easily customize the workflow by adding our own tools for special treatments.

A step-by-step tutorial of the methods used to generate our mouse limb atlas is at <http://kardon.genetics.utah.edu>. Our tutorial includes links to general tutorials describing the use of each tool we mentioned. A script for rendering confocal data in Maya, as well as a prototype spindle-shaped muscle, is available for download from the tutorial. A video demonstrating our workflow is at <http://youtu.be/g61C-nia9ms>.

Reference

1. J. Liu, J. Sun, and H.-Y. Shum, "Paint Selection," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 69.

Segmentation

Segmentation is the process by which individual structures (for example, the semimembranosus muscle) to be modeled are identified and outlined. For the mouse limb atlas, we had to identify and segment individual tendons, muscles, nerves, and bones. We identified these structures in the CLSM images. CLSM generates high-resolution scans along the xy -plane, with the desired structures separated into different channels. However, the z resolution is often much less than the resolution along the xy -plane (see Figure 1). For the limb data, the voxel aspect ratio is 1:1:16, so the volumetric dataset from CLSM is anisotropic. This makes segmentation of most structures easier

from the xy slices than any other orthogonal or oblique direction.

Structures in the same confocal channel (that is, muscles, nerves, or tendons) usually have similar shapes. These shapes determined which tools and methods we chose to segment the structures.

Muscles

Segmenting muscles is difficult with automatic algorithms because boundaries between densely packed muscles are often obscure. The fine fibers in each muscle further complicate this.

We mainly used Photoshop's quick-selection tool, a semiautomatic tool for feature selection on 2D images (see Figure 2). First, we loaded the xy

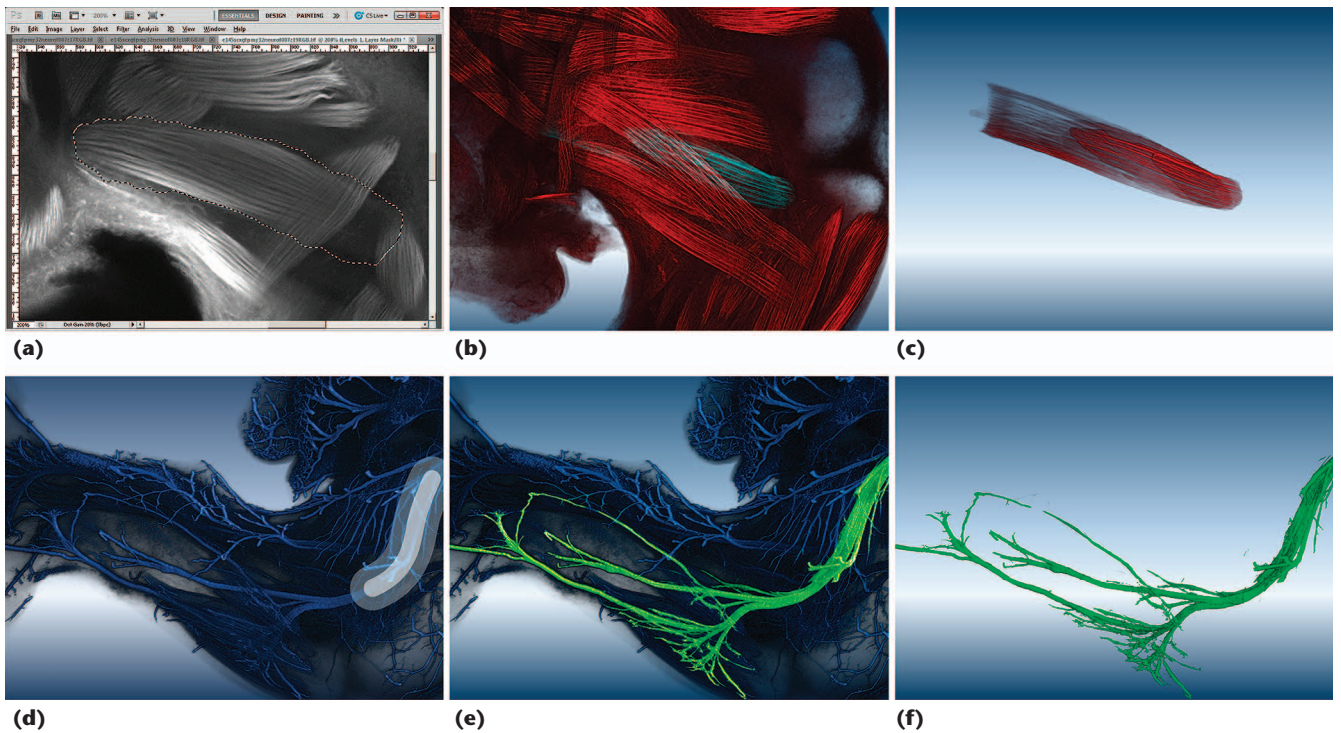


Figure 2. Segmentation using Photoshop (top row) and FluoRender (bottom row). (a) We loaded the xy image slices of the muscle channel into Photoshop. We used the quick-selection tool to select the semimembranosus muscle. We then repeated the process for multiple slices. (b) The selection is visualized with volume rendering. (c) The selection is visualized by isolating it from other structures. (d) We loaded the nerve channel into FluoRender and then painted on the rendering result with FluoRender's selection tool (the white stroke). (e) We selected a bundle of nerves. (f) We extracted the selected nerves.

image slices into Photoshop as individual documents. Photoshop provides a function to match the zoom and location of all opened documents, which let us easily browse through the slices without losing track of a certain feature. Then, we added an adjustment layer on top of the slice image. For feature enhancement and brightness and contrast adjustment, it's sufficient to use the level adjustment, which provides controls for an image's shadows, highlights, and gamma. In our case, we wanted to brighten the signal and increase the image contrast, so we adjusted the highlight value and gamma. Most of Photoshop's adjustment functions are applied only to the displayed result, without changing the original image. So, we could adjust the values for different regions of the slice during selection.

As Figure 2 shows, we then used the quick-selection tool on the image slice to select a muscle. The quick-selection tool tries to find prominent structural boundaries. We could fine-tune the selected result with a smaller brushstroke. However, precise segmentation wasn't only impractical but also unnecessary, because we subsequently imported the segmented results into a modeling tool (Maya) and visualized them with volume rendering. A human modeler can then build high-quality polygon models according to the volume-rendered

results, which can visualize structural boundaries in 3D. So, in our workflow, segmentation aims mainly to reduce complexity for easier modeling.

Nerves

Owing to nerves' complex structures in 3D, separating them from background noise can be challenging. The muscle segmentation method, as well as similar methods that require selections on a 2D plane, is ineffective. Applying a global transfer function to suppress the noise can't effectively handle delicate and branching structures, especially those deeply buried in other tissues.

To extract the nerves, we used FluoRender's volume paint selection tool. Similarly to Photoshop's quick-selection tool, this tool operates directly on the volume-rendered results, employing user-guided diffusion to find structural boundaries. Figure 2 shows the result of nerve extraction. We used the same technique to reduce the complex structures of the limb nervous system into small groups. Then, less occlusion occurred when we rendered individual groups for modeling.

Tendons

Tendons attach muscles to bones. The tendons' shapes can vary from simple cylinders to complex branched structures. Complex, branched tendons

are more easily selected in 3D, so we used FluoRender's volume paint selection tool to select and segment them. For simpler, unbranched tendons, we used Photoshop.

Bones

As we mentioned before, bones appeared as black regions (unlabeled by the fluorescent tags) in the green and red channels. They can't be easily visualized by volume rendering. However, their relatively simple shapes let us easily segment them in Photoshop using the same method as for muscles.

Modeling

Modeling is the process of building a polygon model for each structure (for example, individual muscles, nerves, tendons, and bones). Previous digital anatomical atlases for biological research have used automatic algorithms or programs to generate polygon models directly from segmented data.^{1,2} Although the mesh quality has greatly improved from a computational viewpoint, CG artists still consider automatically generated polygons to be of low quality. First, most automatic algorithms have difficulty generating a quad mesh, which is considered the basis for good model structuring. Second, polygon placement is inefficient. The polygon contours often don't follow meaningful structures of the segmented volume. Instead, they usually congregate at noisy regions and form distracting patterns. Third, automatic algorithms often generate high polygon density, which makes further manual adjustment difficult, if not impossible. Finally, there's no algorithm that can handle complicated shapes and generate high-quality models, such as those we see in the nervous system.

We prioritized quality over automation, so we used manual modeling. In the future, we could increase efficiency by automating some of this process.

After using Photoshop or FluoRender to save the segmentation results as stacks of grayscale images, we imported the stacks into Maya. We then built a coarse polygon model for each structure (for example, individual muscles, nerves, tendons, and bones) to fit the segmentation results' shapes. We converted the coarse polygon model to Maya's subdivision surface model because we could still easily manipulate different levels of details even after smoothing the coarse model through subdivision. After biologists confirmed the subdivision surface model's shape, we converted it to the final polygon model.

We focus here on building the coarse polygon model because the smoothing process via Maya's subdivision surfaces is trivial. For step-by-step instructions on the workflow, visit our tutorial (for more on the tutorial, see the "Our Tools" sidebar).

The capability of visualizing volumetric data with a polygon-modeling tool was crucial to building our atlas. Maya doesn't directly support importing and rendering volumetric data other than procedurally generated 3D textures. So, using Maya's embedded language, we wrote a script that implements a slice-based, real-time volume renderer in Maya's user interface. We then classified the segmented structures' shapes into six categories: spindle shaped, flat, spherical, tubular, branching, and irregular. For each category, we chose slightly different modeling functions to make the coarse polygon model.

Spindle Shaped

Most limb muscles and tendons are spindle shaped. We modeled a prototype muscle, which had a spindle shape and just enough vertices for easy reshaping. First, we translated, rotated, and scaled the prototype model to match the imported volume's general shape. We then tweaked the prototype model's vertices to closely match the imported volume. For certain muscles and tendons, we attached small substructures to the spindle shape. We could easily add these details by extruding faces of the prototype model.

Flat

Some muscles are flat and triangular. Similarly to the modeling method we just described, we created a box (instead of the spindle-shaped prototype model) and fit it to the muscle shape. Shaping the model of a flat muscle required additional adjustments to the vertices.

Spherical

Some foot bones have spherical shapes. Modeling these structures was similar to modeling the flat ones: we created a box and fit it to the shape.

Tubular

These structures differ from spindle-shaped structures by either being longer (for example, the semitendinosus muscle), which made the number of vertices of the prototype model insufficient, or featuring extra structures at the ends (for example, most bones). We created a box either at one end or in the middle of the structure, and then progressively extruded the model faces until the whole structure tightly fit.

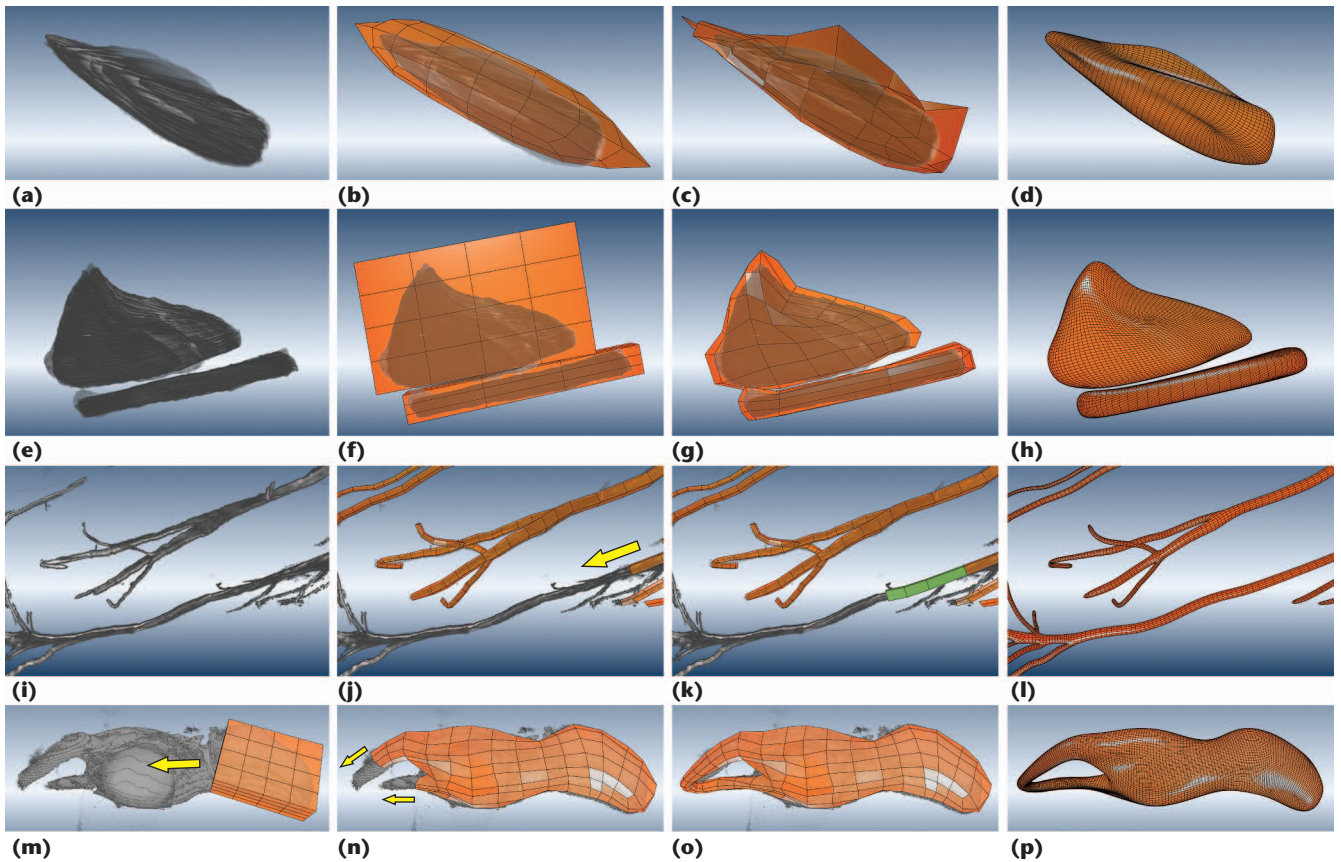


Figure 3. Modeling. (a) The tibialis anterior muscle is spindle shaped. (b) We loaded the prototype muscle to match the muscle's general shape. (c) We tweaked the prototype muscle's vertices to form the coarse model. (d) We smoothed the coarse model. (e) The biceps femoris muscle has two parts, one flat and the other tubular. (f) We created two boxes. (g) We tweaked the boxes' vertices. (h) We smoothed the coarse models. (i) We loaded the nerves into Maya. (j) We modeled the nerves by extruding polygon faces, as indicated by the yellow arrow. (k) The model's green faces are the extruded geometries. (l) We smoothed the completed nerve model. (m) The pelvis has an irregular shape. We first created a box that partially covered the structure. (n) We modeled the box and extruded selected faces of it in the direction indicated by the yellow arrows. We modeled the structure's irregular part similarly to branching structures. (o) The two branches join to form the opening. (p) We smoothed the coarse model.

Branching

As in segmentation, branching structures can cause problems for modeling. Depending on the intended level of detail, branching structures, especially the nerve models, can reach a complexity of 10^4 vertices even for coarse models. Manipulating the vertices takes much time.

However, our modeling technique wasn't so different from that for tubular structures. We extruded selected faces, which then formed a tubular structure in each branch. It was important to recognize regions where branching began and leave enough polygon faces for extruding.

Irregular

Any structure that doesn't fit in the previous categories is irregular. For example, the pelvis has an opening at one end. We used a combination of the previous methods to model irregular structures because we could decompose their shape into several nearly regular shapes. First, we created a

box covering a regular-shaped part of the structure. We then subdivided the box to increase the number of faces, which we progressively extruded to form the irregular shape.

A Modeling Example

Figure 3 illustrates the modeling of several different-shaped structures. Most of the structures had regular shapes, which were easy to model. From our experience, human modelers are good at distinguishing between biological features and noise, even when they aren't familiar with the actual anatomy. This is crucial for modeling confocal data because the high anisotropy can often cause automated modeling to produce spurious step-shaped features.

Texturing

Texturing is the process of creating a surface texture for the model of each structure. Previously, most atlas-building projects in biological research

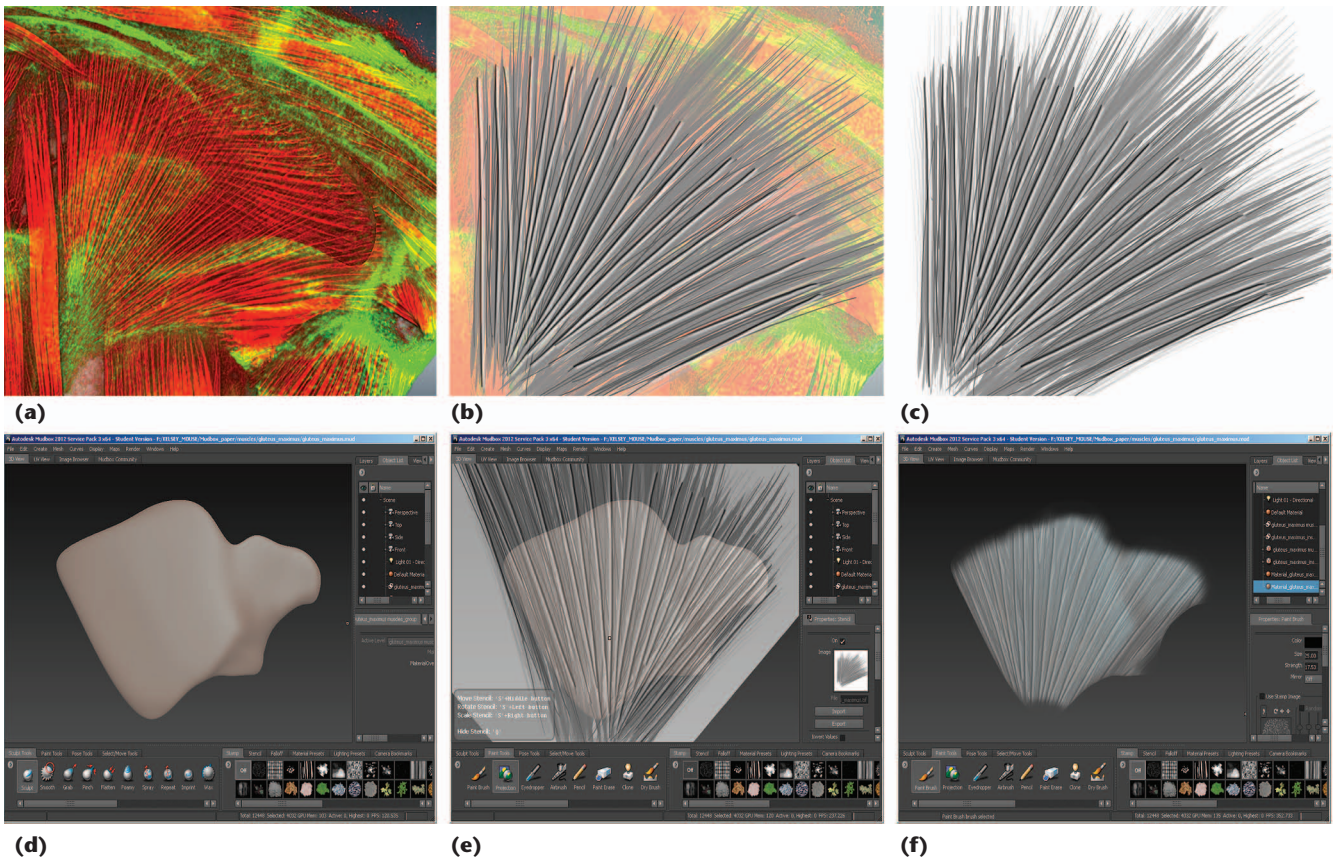


Figure 4. Texture painting. (a) We loaded the rendering of the gluteus maximus muscle into Photoshop. (b) We generated patterns according to the volume-rendered result. (c) The image served as a stencil. (d) We loaded the polygon model of the muscle into Mudbox. (e) After loading the stencil, we used Mudbox’s projection brush to paint the stencil onto the model. (f) We transcribed the stencil image onto the model.

concluded after building the polygon models. Polygon models by themselves have good shape representations of the structures but lack the details defining certain anatomical features, such as the nerves, we skipped manual stitching owing to the high complexity. When we smoothed the coarse polygon models through subdivision surfaces, the UVs were automatically interpolated and required no further editing.

We exported the finished polygon models as separate files in the Wavefront OBJ format and imported those files into Autodesk Mudbox for texture painting (see Figure 4). Instead of painting directly onto the models, we generated stencil patterns in Photoshop and projected the patterns onto the polygon models with Mudbox’s projection brush.

Before we could apply textures, we had to create texture coordinates (UVs, where u and v are the texture space coordinates) of the polygon models, flatten those coordinates, and map them into a unit square of the texture space. The process is called *UV unwrapping*. We unwrapped the UVs before smoothing the coarse models. For spindle-shaped structures, we used a cylindrical projection to generate the UVs of the prototype model. So, for structures built from the prototype model, UVs needed no further editing. For structures of other shapes, we used Maya’s automatic UV projection, which usually generates separate pieces of

UVs. For better texturing quality, we then manually stitched the UV pieces for models of regular shapes. However, for branching structures such as the nerves, we skipped manual stitching owing to the high complexity. When we smoothed the coarse polygon models through subdivision surfaces, the UVs were automatically interpolated and required no further editing.

We exported the finished polygon models as separate files in the Wavefront OBJ format and imported those files into Autodesk Mudbox for texture painting (see Figure 4). Instead of painting directly onto the models, we generated stencil patterns in Photoshop and projected the patterns onto the polygon models with Mudbox’s projection brush.

To generate clear, illustrative patterns, we referenced both the volume-rendered results and the textures from anatomy books. For example, for the gluteus maximus muscle, we first captured its volume rendering and imported it into Photoshop. Then, we painted the muscle fiber pattern by referencing the fiber direction of the volume rendering. Most similarly structured models, such as the spindle-shaped models, could share a stencil

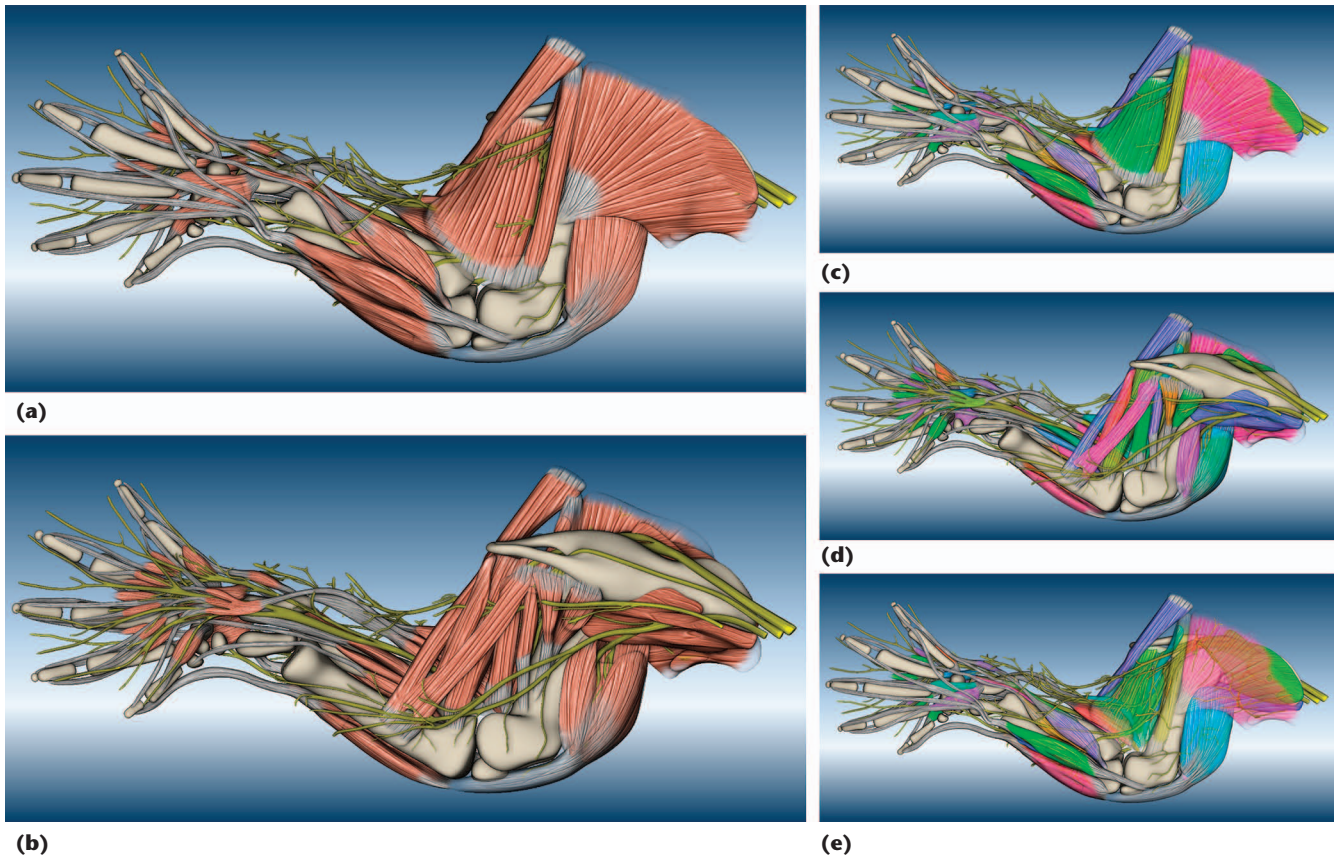


Figure 5. Examples from the limb atlas. (a) The lateral side. (b) The medial side. (c) The lateral side, with muscles alternatively colored. (d) The medial side, with muscles alternatively colored. (e) The biceps femoris and gluteus maximus muscles have increased transparency, revealing underlying structures.

pattern. We used only grayscale in the stencil patterns because we could tint the final textures with arbitrary colors.

Texture painting with projection is intuitive with Mudbox, even for users with no formal training in painting. Different from the projective texture placement that many other programs support, Mudbox's projection brush provides versatile local adjustments for projecting the stencil pattern. During painting, users can independently rotate and translate the pattern image and the polygon model. We also used Mudbox's stamp-based painting brush to apply textures on complex structures such as nerves. Such structures require simple patterns for enhanced realism but can't be easily handled with texture projection.

Presentation

We exported the finished atlas as individual files in the OBJ format, which we could easily convert to many other polygon model formats if needed. The individual models can be assembled and organized with a variety of model-viewing tools.

For interactively viewing the atlas, we used FluoRender, which we used to generate the final renderings in Figure 5. Because FluoRender supports

rendering semitransparent polygon models with depth peeling,³ users can easily adjust the structures' transparency and focus while maintaining an informative context. As a volume-rendering tool, FluoRender let us simultaneously view the raw data and the polygon models.

Evaluation

As we've shown, our workflow can generate high-quality textured polygon models. However, the workflow seems to require much manual work. So, many biologist readers will probably still ask, "How easy is it to carry out? How much time does it take?"

Creating the atlas took four months. The time included harvesting, processing, staining, and scanning of the biological sample, which only experienced biological researchers can do. For the remaining workflow, we collaborated with the students of CS3650,⁴ a digital-character-production course jointly offered by the University of Utah's Department of Film and Media Arts and School of Computing. Two student volunteers participated in the workflow, and their work partially contributed to the course requirements. Finishing the atlas took an academic semester, with all participants working only part-time on it.

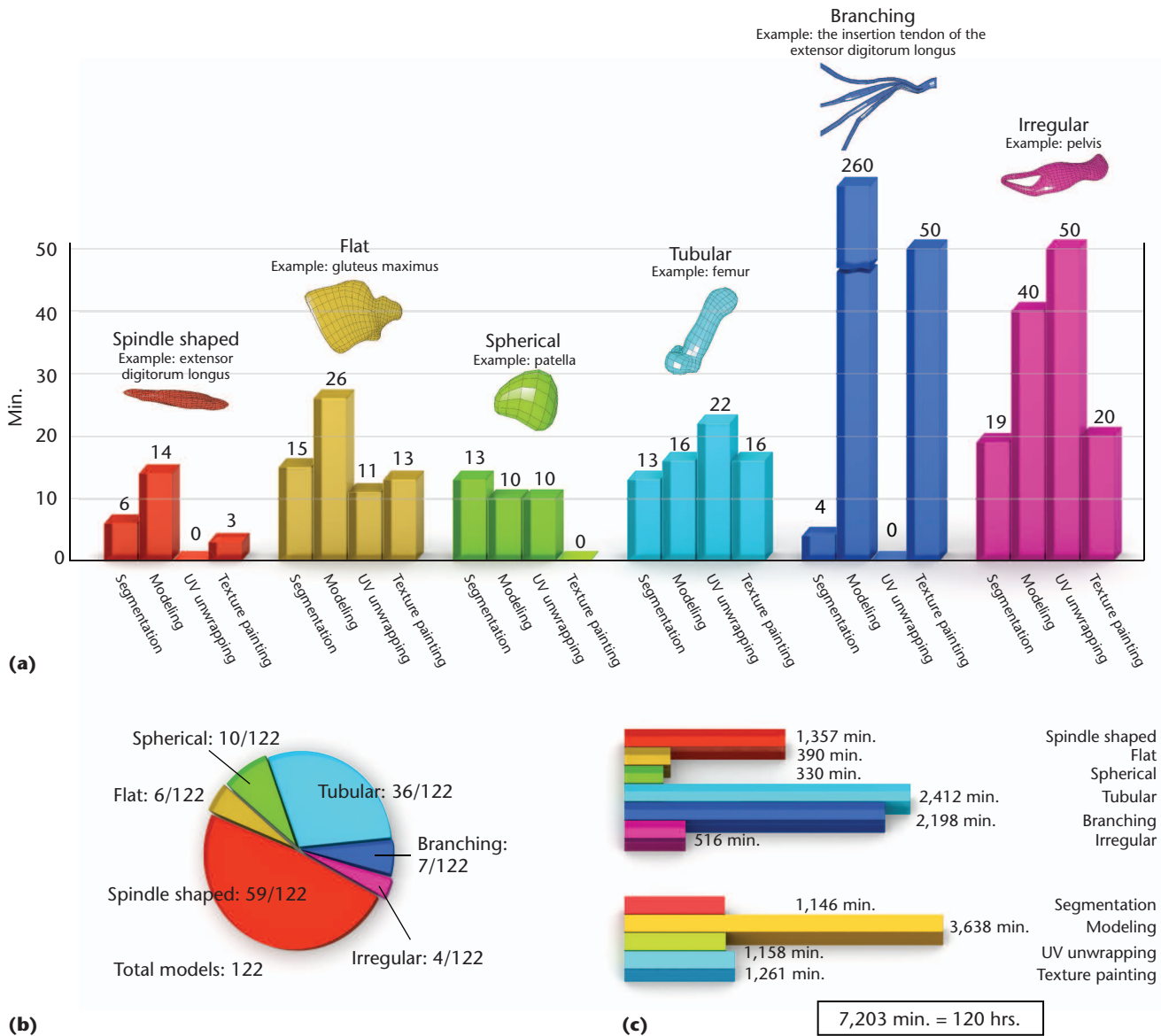


Figure 6. The time spent making the atlas. (a) The average time to construct one model for each of the six categories. (b) The number of models we made for each shape category. (c) The time spent for all 122 models. Building the atlas took approximately 120 hours. We calculated time as the difference between the creation and last modification time stamp of each file. Whenever such information was unavailable or considered inaccurate, participants estimated their time.

Figure 6 illustrates a detailed analysis of the work required for segmentation, modeling, UV unwrapping, and texture painting. Because we started with a prototype model, the spindle-shaped models, which constitute almost half of the atlas, took the least amount of time on average. Regular-shaped (flat, spherical, and tubular) models took less time than complicated structures, which required much time for planning polygon outlines and manipulating vertices or UVs.

On the basis of this analysis, we estimate a project similar to ours will take two weeks to one month, with two persons working full time. For building a different anatomical atlas, the makeup

of the model shapes might change, but their classification should still be valid.

Our workflow is efficient for building high-quality polygon-based anatomical atlases in a relatively short time period. With training in Photoshop, Maya, Mudbox, and FluoRender, biologists can carry out this workflow themselves. (For details on training and learning 3D modeling, see “Digital Visualization Tools Improve Teaching 3D Character Modeling.”⁴)

However, we believe a project such as this is best carried out by an interdisciplinary team. Biologists might find applications other than atlas building for the skills they learn from using the artists’ tools.

Artists will learn anatomy and practice accurate modeling based on real biological data. Computer scientists will gather valuable experience with biological modeling, which they can use to design effective programs to automate manual operations such as fitting polygon models and generating textures.

The finished atlas can be used for building models of a similar biological species or a different developmental stage of the same species, where the coarse polygon models can be easily manipulated and textures reused. The workflow and chosen tools can be applied to most volumetric data acquired from imaging methods other than CLSM.

With researchers being able to create anatomical atlases with readily available tools, the number of available atlases can increase rapidly, and a free, publicly accessible database of atlases can exist. Such a database will be helpful for students, doctors, researchers, medical illustrators, biological-visualization experts, and digital artists. In addition, we hope our research raises the awareness of using artists' tools for scientific research and promotes interdisciplinary collaborations between artists, computer scientists, and biologists. ■■

Acknowledgments

Yong Wan and A. Kelsey Lewis contributed equally to this work. This research is supported by King Abdullah University of Science and Technology award KUS-C1-016-04; the US Department of Energy Scientific Discovery through Advanced Computing Visualization and Analytics Center for Enabling Technologies and Institute of Scalable Data Management, Analysis, and Visualization; US National Science Foundation grant OCI-0906379; and US National Institutes of Health grants NIH-1R01GM098151-01 and R01HD053728 NICHD. We thank Ronen Schweitzer of Oregon Health & Science University for the Scx-GFP mice; Chris Rodesch, director of the University of Utah's Fluorescence Microscopy Facility; Chems Touati of the University of Utah's Scientific Computing and Imaging Institute for making the demo video; and student participants Hongyuan Li and Damean Lyon.

References

1. A. DeLaurier et al., "The Mouse Limb Anatomy Atlas: An Interactive 3D Tool for Studying Embryonic Limb Patterning," *BMC Developmental Biology*, vol. 8, 2008, article 83.

2. T. Ju, "Building a 3D Atlas of the Mouse Brain," PhD thesis, Dept. of Computer Science, Rice Univ., 2005.
3. C. Everitt, "Interactive Order-Independent Transparency," white paper, Nvidia, 1999.
4. M. van Langeveld and R. Kessler, "Digital Visualization Tools Improve Teaching 3D Character Modeling," *Proc. 41st ACM Tech. Symp. Computer Science Education (SIGCSE 10)*, ACM, 2010, pp. 82-86.

Yong Wan is a PhD student at the University of Utah's Scientific Computing and Imaging Institute and works on the FluoRender project. His research interests include confocal visualization and 3D modeling. Wan has a BS in electronic engineering from Southeast University, China. Contact him at wanyong@cs.utah.edu.


A. Kelsey Lewis is a lab technician in the University of Utah's Department of Human Genetics. Her research interests include developmental biology, evolutionary biology, and imaging techniques. Lewis has a BA in biology from Mount Holyoke College. Contact her at aklewis@genetics.utah.edu.

Mary Colasanto is a PhD student in the University of Utah's Department of Human Genetics. Her research interests include the development of the vertebrate limb's musculoskeletal system. Colasanto has a BS in biochemistry from Old Dominion University. Contact her at mcolasanto@genetics.utah.edu.

Mark van Langeveld is an assistant professor in the University of Utah's School of Computing and is the director of the Entertainment Arts and Engineering Master Games Studio Graduate Program. His research interests include digital content creation and teaching methods in computer graphics. Van Langeveld has a PhD in computing from the University of Utah. Contact him at longfieldstudio@gmail.com.

Gabrielle Kardon is an associate professor in the University of Utah's Department of Human Genetics. Her lab conducts research on musculoskeletal development, regeneration, and diseases. Kardon has a PhD in developmental biology from Duke University. Contact her at gkardon@genetics.utah.edu.

Charles (Chuck) Hansen is a professor of computer science and an associate director of the Scientific Computing and Imaging Institute at the University of Utah. His research has covered scientific visualization, computer graphics, parallel computation, and computer vision. Hansen has a PhD in computer science from the University of Utah. He's an IEEE Fellow. Contact him at hansen@cs.utah.edu.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.