

A Lightweight Tangible 3D Interface for Interactive Visualization of Thin Fiber Structures

Bret Jackson, *Student Member, IEEE*, Tung Yuen Lau, David Schroeder
Kimani C. Toussaint, Jr., *Senior Member, IEEE*, and Daniel F. Keefe, *Senior Member, IEEE*

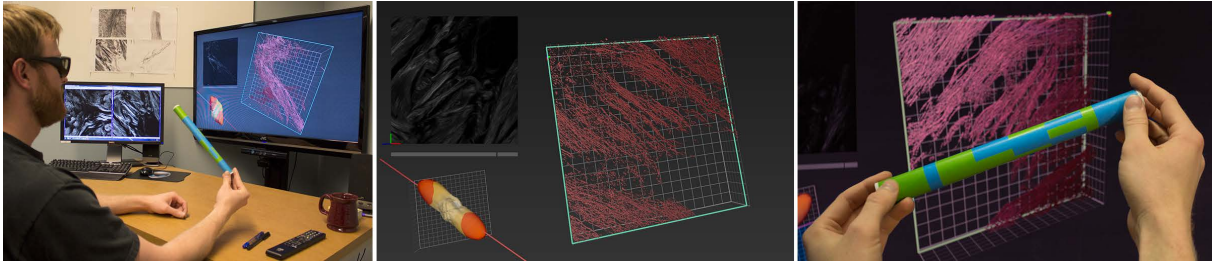


Fig. 1. Left: Exploring fiber orientations in tissue using a paper prop and a commodity VR display. Middle: Linked views show: (1) a stereoscopic rendering of fibers; (2) a 3D fiber orientation histogram; and (3) 2D image slices. Note how only fibers oriented in the direction specified by the prop are rendered. Right: Patterns printed on the prop enable tracking of rolling and other gestures to provide a tangible 3D interface for the visualization.

Abstract—We present a prop-based, tangible interface for 3D interactive visualization of thin fiber structures. These data are commonly found in current biomedicine datasets, for example second-harmonic generation microscopy of collagen fibers in tissue. Our approach uses commodity visualization technologies such as a depth sensing camera and low-cost 3D display. Unlike most current uses of these emerging technologies in the games and graphics communities, we employ the depth sensing camera to create a fish-tank stereoscopic virtual reality system at the scientist’s desk that supports tracking of small-scale gestures with objects already found in the work space. We apply the new interface to the problem of interactive exploratory visualization of three-dimensional thin fiber data. A critical task for the visual analysis of these data is understanding patterns in fiber orientation throughout a volume. The interface enables a new, fluid style of data exploration and fiber orientation analysis by using props to provide needed passive-haptic feedback, making 3D interactions with these fiber structures more controlled. We also contribute a low-level algorithm for extracting fiber centerlines from volumetric imaging. The system was designed and evaluated with two biophotonics experts who currently use it in their lab. As compared to typical practice within their field, the new visualization system provides a more effective way to examine and understand the 3D biomedicine datasets they collect.

Index Terms—Scientific visualization, 3D interaction, tangible interaction, microscopy visualization

1 INTRODUCTION

Natural user interfaces, and in particular spatial (3D) user interfaces, are becoming an increasingly important area of research within the scientific visualization community [13]. Although input technologies, such as multi-touch sensors and depth-sensing cameras, are not new inventions, these technologies have sparked a great deal of excitement recently due to the now widespread availability of low-cost quality sensors in this style, a development fueled in large part by recent applications to the games and entertainment industries.

Scientific visualization applications have great potential to benefit from these advances, including the emergence of low-cost 3D TV

displays. For example, it is now quite reasonable to expect that a biologist, doctor, or other scientific researcher can work with a desktop-scale head-tracked stereoscopic display (fish-tank virtual reality) directly in his/her office. Just a few years ago, such a hardware system would have been prohibitively expensive and/or complex to maintain for personal data visualization. Given this context, our research addresses the important challenge of developing effective 3D user interfaces for desktop-scale stereoscopic data visualization. In particular, we aim to rethink the way that we interact with depth-sensing cameras, such as the Microsoft Kinect, to make this style of user interface more appropriate for accurate, real-time exploration of volume data.

The specific work described in this paper is based on visualizing thin fiber structures from biomedicine data (Figure 1). The system was developed in collaboration with two biophotonics experts that use second-harmonic generation (SHG) microscopy to study the effects of tendonitis on collagen fiber organization in tissue [22]. We know from our collaborators that analyzing these data currently requires collecting image data and subsequently choosing the appropriate algorithm that optimizes sensitivity to detecting tendonitis and computational cost. This process is repeated, sometimes dozens of times, at the expense of depletion of resources (e.g., tens of hours of labor, hardware use), until the best site of interest of the specimen is chosen for study.

In contrast, in our vision of the future, scientists acquire a volumetric SHG image via real-time streaming to a low-cost 3D visualization system. They then explore the data from multiple vantage points, interactively select regions of interest, query underlying multidimensional data values, fit spatial data to models, and adjust visualization

- *Bret Jackson is with the University of Minnesota, USA. E-mail: bjackson@cs.umn.edu.*
- *Tung Yuen Lau is with the University of Illinois Urbana-Champaign, USA. E-mail: lau20@illinois.edu.*
- *David Schroeder is with the University of Minnesota, USA. E-mail: daschroe@cs.umn.edu.*
- *Kimani C. Toussaint, Jr. is with the University of Illinois Urbana-Champaign, USA. E-mail: ktoussai@illinois.edu.*
- *Daniel F. Keefe is with the University of Minnesota, USA. E-mail: keefe@cs.umn.edu*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

parameters. Based on the insight generated through this process, they re-adjust the SHG microscope (e.g., focus, scan angle, zoom) from within the interactive system to look for specific queues in the collagen fiber organization that are highly correlated with the onset of tendonitis, queues that may not have been easily revealed or remain undiscovered using today’s imaging.

Working toward this vision, this paper presents the design and implementation of a low-cost stereoscopic data visualization system for analyzing collagen fibers captured via SHG microscopy. In addition to the discussion of the complete system and user feedback, our main technical contribution is a tangible user interface based on depth-camera technology. Following the motivation described above, we intend for this interface to be usable at a scientist’s desktop. Therefore, the tangible props used in the interface are simply constructed from paper printouts (see Supplemental Material). We argue, based on related research results in the 3D user interfaces community (e.g., [7]), that using passive haptic props dramatically increases the control and understanding that users have when working with 3D interfaces and data. Our work contributes a demonstration of how this can be achieved using today’s hardware, while maintaining a “lightweight” interface in the sense that scientists are not required to buy a kit of plastic blocks [18] or generate a 3D rapid prototype object [17]. To realize the full-featured visualization system, we also contribute an algorithm for extracting thin, dense fiber features (centerlines and diameters) from volumetric data and a real-time stereoscopic rendering strategy for this type of fiber data.

2 RELATED WORK

2.1 Passive Haptic 3D User Interfaces

It is now well established in the 3D user interfaces and human-computer interaction communities that even passive haptic props typically provide great advantages over “freehand” interfaces in terms of spatial understanding and control over 3D operations. Hinckley’s early work in this area, also applied to scientific visualization, used simple props tracked in space (a doll’s head, a clear square of plastic) to help doctors fluidly explore and slice through brain imaging data [7]. Accurate physical 3D data printouts from rapid prototyping machines have also been used as interaction props for 3D data visualization [11, 17]. Touch-sensitive display interfaces also share some similarity to our work in that they enable fluid, natural styles of interaction with data [14], and research has shown that the passive haptic feedback of placing one’s fingers and/or hand against a surface facilitates accurate gestural interaction with data [5, 12]. Despite this evidence, most of today’s interfaces based on depth-sensing cameras utilize full-body tracking and rough gesturing in the style commonly found in games [24]. Although we argue that this style of interaction is less appropriate for scientific visualization applications, it does have the advantage that it is “lightweight”. Scientists can walk right up and use these interfaces, and they do not need to invest time and money into custom rapid prototype props or costly 3D tracking systems. Our work aims to maintain the spirit of a natural, lightweight interface while also gaining the advantages of passive haptic feedback. One way that we achieve this is by using simple props that are easily constructed from printed paper, which is already available in an office work space.

2.2 Visualization of Thin Fiber Structures

Scientific applications often require visualization of collections of 3D curves through a volume; both fluid flows (e.g., streamlines) and DT-MRI data (e.g., neural fiber tracts) require visualization of fiber-like data. Many graphics algorithms have been developed to depict these structures. We believe some of the most effective are the algorithms that strive to enhance the user’s perception of depth and fiber crossings. A popular method for achieving this is rendering a “halo” around each fiber so that when two fibers cross each other their colors do not blend together. The frontmost fiber’s halo (usually drawn in the same color as the background of the scene) occludes any fibers that are farther away from the viewer, making the frontmost fiber stand out as on top of the others. This has been implemented in volume renderings [10, 25] and, more recently, in GPU shaders [6]. Our rendering strategy

builds upon the recent work of Everts et al. to also include perceptually motivated coloring, lighting, and stereoscopic rendering.

2.3 Feature Detection in Bioimaging Data

Visualization of fiber orientation in biological tissue is important for disease assessment [3]. Our visualization system is designed to work with orientation data generated from bottom-up techniques which identify the orientation of individual fiber centerlines. A variety of techniques in this style exist; however, the scattered-snakelet approach [23] is appealing because it quantifies the orientation of short sections of each fiber. These short sections, called snakelets, can easily be compared with a user-specified vector to highlight fiber sections with a similar orientation. We introduce an adaptation of the scattered-snakelet approach that improves fiber segmentation and snakelet merging. This allows the technique to be applied to dense fiber tissues. Unlike other published methods for centerline extraction from dense tissues, the modified scattered-snakelet approach is better able to handle fiber crossings or fibers with uneven surfaces because it does not rely on thinning [16] or recursively growing the centerline [26].

3 A LIGHTWEIGHT TANGIBLE 3D USER INTERFACE

As shown in Figure 1, the system hardware consists of a 3D TV display, a depth-sensing camera placed in front of the display, and a simple paper prop held by the user. The following sections describe the key features and algorithms of the system. The workflow begins with a feature extraction step, which identifies features in the volumetric image stack and exports a set of fibers for visualization. The visualization system then imports these data to be displayed on the 3D TV.

From a scientific standpoint, the most critical aspect of these data to understand is the geometric alignment of fibers. For example, fibers in injured tendons are oriented in various directions, unlike normal tendons which exhibit highly aligned fibers [22]. So, understanding the alignment of fibers could lead to better diagnosis of tendonitis.

The prop design is motivated by our observations of how people naturally gesture when discussing these fiber data in front of a traditional 2D display. Even as we discussed the data within our own research group, we found ourselves naturally using the cylindrical shape of a pen as a handheld prop for indicating a 3D orientations. Although we discovered that current depth-camera technology was unable to robustly track the small diameter of a pen, the slightly larger diameter of a rolled piece of paper met the requirements that the prop be: (1) literally lightweight; (2) readily available in an office; and (3) contain a long axis for indicating a direction. Additionally, working with a rolled piece of paper makes it possible for a pattern to be easily printed on the surface to facilitate tracking and additional interactions.

The prop interface supports three critical interaction tasks for exploratory visualization: (1) indicating a 3D vector to query the data, (2) setting scalar values, and (3) reorienting the 3D volume.

3.1 Gestures for Indicating a 3D Vector

Indicating a 3D vector is a common function when analyzing fiber data. A 3D prop-based approach provides an immediacy that is lacking in traditional 2D mouse-based methods. Additionally, the prop serves as a physical display of the vector orientation, which is advantageous when collaborating and conveying spatial information to others.

To indicate a 3D vector, the prop is held with one hand in a pose that users naturally adopt when discussing the data. The orientation of the prop in 3D space is used to control the visibility of fibers, as shown in Figure 2. Fiber segments that align (within some similarity threshold) with the same orientation of the prop are displayed, while all other fiber segments are made completely transparent. The *orientation similarity threshold* used in this operation is simply a scalar value between 0 and 1. To determine whether fiber segments match the orientation of the prop, the orientation vector for the long-axis of the prop is compared to the orientation of each fiber segment; if the absolute value of the dot product of these two vectors is greater than the value for the orientation similarity threshold, then the corresponding fiber segment is displayed. The result is that the user is able to fluidly and quickly explore a fiber dataset to achieve an understanding of the predominate

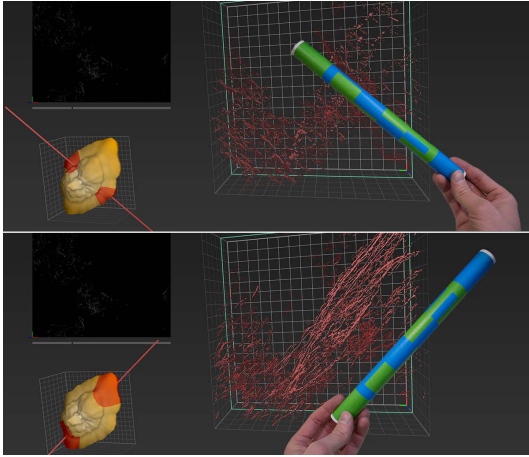


Fig. 2. Holding the prop in one hand will show fibers oriented within a similarity threshold of the props orientation.

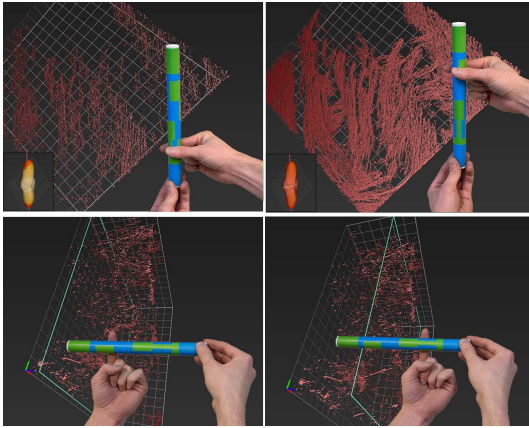


Fig. 3. Top: Holding the prop with one hand and sliding the other up or down the prop will set the similarity threshold. Bottom: Sliding horizontally adjusts the position of the 2D image slice that is displayed.

fiber orientations and their spatial distribution within the volume. The accompanying video demonstrates the fluidity of this interaction.

3.2 Gestures for Setting Scalar Values

The orientation similarity threshold described above is one example of a scalar value that needs to be set interactively. Since setting scalar values is a common operation in any visualization system, we judged it to be important to also support this action with the prop-based interface so that the user can set these values immediately while working with the interface, i.e., without needing to first put down the prop.

As illustrated in Figure 3, scalar values are set by using the prop like a 2D slider. The prop is held at its end by one hand, while the fingers of the other hand slide along the prop's length in a pinching gesture to indicate a value. Multiple scalar values can be set by holding the prop in different orientations as the user performs the gesture. For instance, if the prop is held vertically the orientation similarity threshold is adjusted, while if it is held horizontally the index of the 2D image slice displayed in the upper left corner of the screen is adjusted.

The user's motion can be interpreted based either on the absolute position of where his/her finger crosses the prop or on the relative change in the sliding motion. For our applications, the orientation similarity threshold is set using the absolute position. The threshold is calculated using a non-linear scale to increase precision at the high end of the scale where only the most similar fiber sections are rendered.

To adjust the index of the 2D image slice displayed in the upper left of the visualization, the user performs a similar gesture with the prop

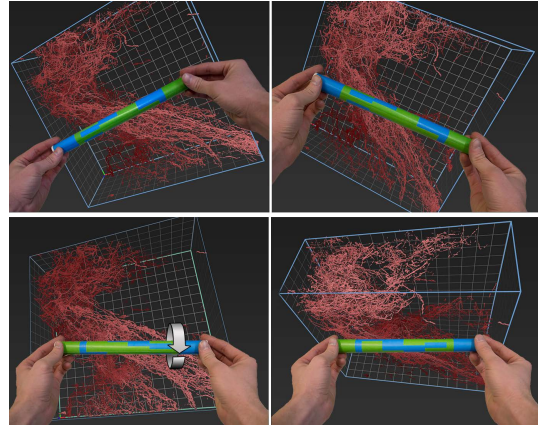


Fig. 4. Holding the prop with both hands will 'grab' the volume, and rotating or rolling the prop will rotate or roll the volume around its center point.

held horizontally. The same method of detection is used. However, it is useful to have more precision in this manipulation than with the threshold adjustment. So, a relative scale is used, and the user must clutch and slide his or her finger more than once along the length of the prop to navigate through an entire stack of about 150 images.

3.3 Gestures for Reorienting a 3D Volume

One of the most critical tasks to support in this visualization is reorienting the 3D volume to investigate the data from different viewpoints. Conceptually, this is similar to the task of indicating a 3D vector in that both tasks involve changing an orientation; however, we think of reorienting the volume as a more forceful action since it moves all of the data displayed on the screen. So, as shown in Figure 4 we use a more forceful grip on the prop (two hands rather than one). To the user, the impression is that his or her hands are grabbing onto the virtual scene and the scene responds to his or her rotational movement as expected. The rotation of the prop from one moment to the next is mapped to a corresponding rotation around the center of the scene. This follows a style typical of virtual reality navigation techniques (e.g., [19]).

The extension in our case is that rather than "grabbing the air", the user has a physical prop to hold. We found that this leads to a natural style of rotation for two of the three axes of rotation that is perhaps even more intuitive than "grabbing the air" due to the physical feedback provided by the prop. However, a control is still needed to rotate along the axis parallel to the long-axis of the prop. To accomplish this, we introduce another gesture – the user simply rolls the prop in his/her hands as shown in Figure 4 (bottom row). Section 4.6 describes how the specific pattern printed on the prop facilitates this interaction.

3.4 Seamless Transitions between Gestures

A final important characteristic of the interface is that it facilitates smooth transitioning between all of the gestures. Rather than explicitly switching the interaction mode with a button press or other discrete input, the design of this gesture set makes it possible to set the interaction mode implicitly based on how the user holds the prop. This type of spring-loaded mode selection has previously been found to reduce mode-based user errors [21]. If the prop is held with one hand, then the gesture for indicating a 3D vector is recognized – this is the most common gesture. If the prop is held with two hands, then the position of the hand grips must be checked. If the hands are located at opposing ends of the prop, then the gesture for reorienting a 3D volume is recognized. If one hand is at an endpoint of the prop and the other is in the middle of the prop, then the gesture for setting a scalar value is recognized, and the orientation of the prop is used to determine exactly which scalar value is being set.

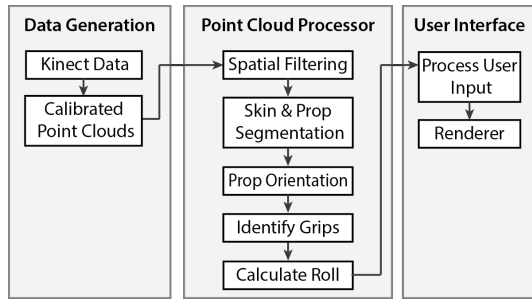


Fig. 5. Overview of the visualization system’s three modules. These modules work in parallel to track the user’s interaction, process the resulting point cloud, and update the rendering based on the input.

4 IMPLEMENTATION

As shown in Figure 5, the visualization system is implemented with three modules that execute in parallel. The data processing module uses the color and depth images produced by the depth camera to create a 3D point cloud of the volume in front of the display. This point cloud is passed to the point cloud processor module, which: spatially filters the data, segments out the hands and prop, determines the prop orientation, identifies whether the prop is held with one or two hands, and tracks the roll gesture. The point cloud processor module is able to process the point data in 10 ms on a quad core 3.4 GHz Intel i7-2600 CPU with 16 GB of RAM. The user interface module responds to updates from the cloud processor and performs the rendering.

4.1 Generating an Accurate 3D Point Cloud

The Microsoft Kinect produces color and depth images at 30 frames a second. These images are combined to project each color pixel to its corresponding 3D position, generating a 3D point cloud of the scene. Because the Kinect’s color and IR camera are spaced apart, there is not a one-to-one mapping between the color and depth image pixels. At close distances, we found a significant misalignment between the color and depth in the factory calibration.

Our solution is to register the two images using a stereo calibration algorithm [9]. A checkerboard pattern is captured with both the IR and color cameras. Corresponding points in the images are used to determine the extrinsic and intrinsic parameters of each camera. Following a standard multi-camera calibration approach [2], the IR camera’s intrinsic parameters (principal point and focal lengths) are used to reproject each 2D depth pixel into the IR camera’s 3D coordinate space. Then, using the extrinsic parameters of the system (the rotation and translation between the cameras), these 3D points are transformed into the color camera’s 3D coordinate system. Finally, the color camera’s intrinsic parameters are used to project the 3D color points to 2D pixel locations in the color image. The result is a 3D point cloud where the correspondence between depth and color values is more accurate.

4.2 Filtering to the Volume of Interest

To improve performance of the point cloud processing, the 3D points are spatially filtered to remove points not in the vicinity of the prop. We assume that the closest point to the Kinect belongs to the user’s hand or the prop. Using this assumption, points that lie further in depth than a band slightly wider than the length of the prop can be removed from the cloud. This removes the user’s head, torso, and the background. Assuming the highest point in the remaining cloud lies on the hand or prop, we also filter vertically based on the length of the prop. Extraneous points on the wrist are filtered by removing points that lie under a quadratic curve.

4.3 Segmenting the Prop and Hands

The pattern printed on the paper prop is designed to enable a robust segmentation between the prop and the user’s hands. The blue and green colors were chosen for their separability from skin colors which

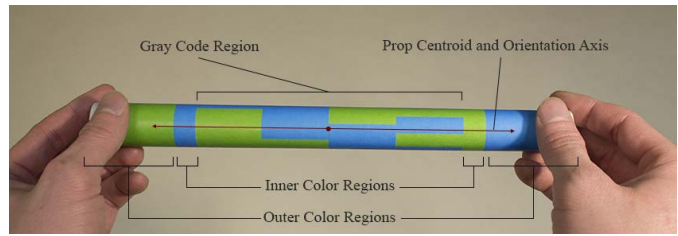


Fig. 6. A specific pattern is printed on the paper prop to support multiple visualization tasks.

frequently exhibit red hues. We found that the results are more accurate than using a black and white pattern, where highlights or shadows on the hand caused frequent mis-classifications as prop points.

Since lighting, skin color, and the particular printer used to create the prop can all impact the color data reported by the camera, we have each new user perform a 30 second calibration routine where the prop is held in several pre-defined poses and the system records the color data observed for each of the three features the algorithm must detect: (1) Skin on the user’s hands or fingers; (2) The blue portions of the prop; (3) The green portions of the prop. The color observations are stored in 2D histograms with 32×32 bins; different hues are arranged along one axis of each histogram and different saturation values are arranged along the other axis, similar to the approach in [8]. After normalizing the histograms, each one can be viewed as a probability distribution. For a color of a particular hue and saturation, the probability that it belongs to the user’s hands is determined by the value in the corresponding hue-saturation bin in the skin feature histogram. This nonparametric model is able to handle the non-linear boundaries between the skin and prop colors, is very quick to compute, and allows the system to be used under various static lighting conditions. Using this strategy, each point in the point cloud is classified as belonging to the user’s hands, a blue portion of the prop, a green portion of the prop, or some other unknown object.

4.4 Detecting the Orientation of the Prop

Once the points have been classified, the prop orientation is calculated using the covariance matrix of the prop points normalized by their 3D centroid. Because the prop’s length is much longer than its diameter, the maximum eigenvector of this matrix indicates the prop’s 3D orientation. From frame to frame, we smooth the value of the orientation vector using the 1€ filter [4], a low-pass filter where the frequency cutoff is based on update speed to balance jitter and lag.

4.5 Identifying Grips

To distinguish between adjusting 3D orientation and other gestures, it is important to detect whether the prop is being gripped with one or two hands. First, the 3D locations of the prop endpoints are calculated. Then, the number of hand points within a 2 cm radius of the prop endpoints is computed. If this number exceeds a preset threshold (15 in our implementation), then a grip is identified at the corresponding end of the prop.

At least one of the user’s hands will always occlude an endpoint of the prop, but since the prop’s geometry and color pattern are known, the 3D location of the endpoints can still be determined. The prop has a solid color band, either green or blue, at each end, followed by a thinner band of the opposing color. (These inner and outer color bands are labeled in Figure 6.) The algorithm projects each point that has been classified as being part of the prop onto the prop’s median axis to create a set of points that lie along a 3D line. Then, it searches for color patterns along this line of points, specifically the algorithm detects the edge between the inner and outer color bands. Once the 3D location of this edge is found, the endpoints of the prop are calculated by translating from these locations outward along the prop’s orientation vector by the length of the outer color band.

4.6 Tracking Roll Gestures

A color pattern was designed for the central portion of the prop to detect the gesture used to reorient the volume by rolling the prop around its median axis. The pattern is a 4-bit Gray code, a cyclical reflected binary sequence where each successive number has a Hamming distance of one. This property makes it easy to detect errors from noisy sensor readings. Because the code is cyclical, i.e. only a single bit changes between the first and last number in the sequence, there is no discontinuity in tracking as the user rolls past the end of the sequence.

To read the current Gray code, we first find the subset of the point cloud points that lie in the Gray code region of the prop. These are calculated in the same manner as described above for finding the edge between the inner and outer color bands, but this time, the inside edges of the inner color bands are detected and only points lying between these edges are used for analysis. These points are filtered to include only those that lie within a 2.5 mm radius of the median axis of the prop, so as to remove any edge points that might pick up color from the surrounding scene rather than the prop itself. The remaining points are divided into four bins according to their 3D positions along the length of the prop. The average color for each bin is calculated and the bin is assigned a 0 or 1 depending on whether the color is green or blue. The pattern is read from the green end to the blue end of the prop to create the Gray code. For example, the Gray code in Figure 6 would be read as 0111.

As the user rolls the prop, successive Gray codes are read and converted to their corresponding binary number. Each increase or decrease in the binary number is mapped to a 5 degree rotation of the fiber volume around the axis indicated by the prop's orientation.

4.7 Detecting Finger Position Along the Prop

During the gestures used to set the orientation similarity threshold and for navigating through the image stack, the prop is used as a physical slider. The points belonging to the sliding finger will be classified as skin. So, to detect the crossing location, the skin points are checked to determine whether they lie within 8 mm of the median axis of the prop and within the length of the prop. The 3D centroid of these points is then calculated and projected onto the prop's axis, and the position of this point along the length of the axis is used to set the appropriate scalar value.

5 REAL-TIME RENDERING

The stereoscopic visualization includes a volumetric display of the fiber data, a 3D fiber orientation histogram, and a number of other visual widgets to facilitate interaction.

5.1 Fiber Rendering

Each dataset consists of short individual fiber segments. A point-based rendering strategy, similar to that introduced by Everts et al. [6], is used to draw each segment; a GPU shader expands points into view-aligned quads, and then applies a depth-dependent halo.

We extend this algorithm to adjust the scale of points based on the fiber thickness and to enhance depth perception using a colormap from light to dark as points recede in depth. The shader is also used to determine the visibility of each point based on whether it lies within the current orientation similarity threshold. Rendering three quads per segment (one per endpoint, and one at the midpoint), the algorithm renders at interactive framerates to a row-interlaced stereo image at a resolution of 1920x1080 on a NVIDIA GeForce GTX590.

Lambertian shading is applied to the quads so that a series of segments appears to be an extruded tube. The surface normals are computed in screen space as if the visible portion of the quad is an infinite cylinder pointing in the direction of the corresponding segment. This makes the cylindrical shape of the fiber data more readily apparent, while the colormap based on depth provides additional depth cues to the user. This double encoding of depth is particularly helpful in areas with few fiber crossings where the depth-dependent halo technique is less effective at conveying depth.

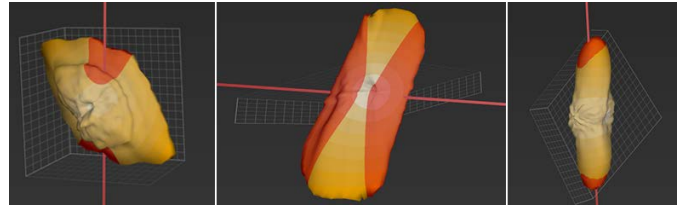


Fig. 7. A spherical histogram depicts the distribution of fiber directions and the subset that is currently displayed based on the current orientation of the prop and the orientation similarity threshold.

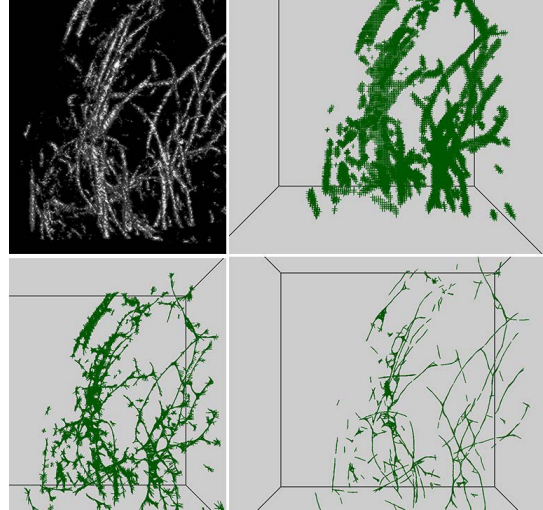


Fig. 8. The Scattered-Snakelet approach for finding fiber centerlines. Top left: Volume rendering of collagen fiber. Top right: Snakelets are initialized on a grid. Bottom left: Snakelet endpoints are moved along the gradient of the distance map. Bottom right: Final centerlines.

5.2 3D Fiber Orientation Histogram

A spherical histogram widget displayed in the lower-left of the visualization conveys the global distribution of fiber orientations within the volume (Figure 7). This widget encodes the orientation data via both shape and a white-to-orange colormap. The shape is elongated and more orange in the directions corresponding with the most prevalent fiber orientations. The spherical histogram and volume are linked to move in tandem when the volume is rotated. A grid drawn behind the histogram reinforces this connection. The prop's current orientation is visualized as a red line in the histogram widget. The similarity threshold is represented by shading all the surface points within that threshold in a red color.

5.3 Additional Visual Widgets

Changes in prop orientation and similarity threshold immediately update both the volume view and the histogram widget. When the user places his/her hands on either side of the prop to reorient the volume, the change in the prop's function is indicated by drawing a blue outline around the bounding grid (Figure 4). A similar blue outline is drawn around the histogram when the user is adjusting the scalar threshold.

To complement the 3D fiber visualization, the 2D microscopy image slices are shown in the upper left corner of the display, illustrated in Figure 9. A single slice is displayed at a time. A 2D bar is displayed below the image to indicate the position of the current slice within the image stack, and the position of the slice is also rendered as a green outline in the fiber volume.

6 IDENTIFYING FIBER STRUCTURES

To generate the data used in the visualization, fiber centerlines are extracted from the volumetric image stack using an extended version of

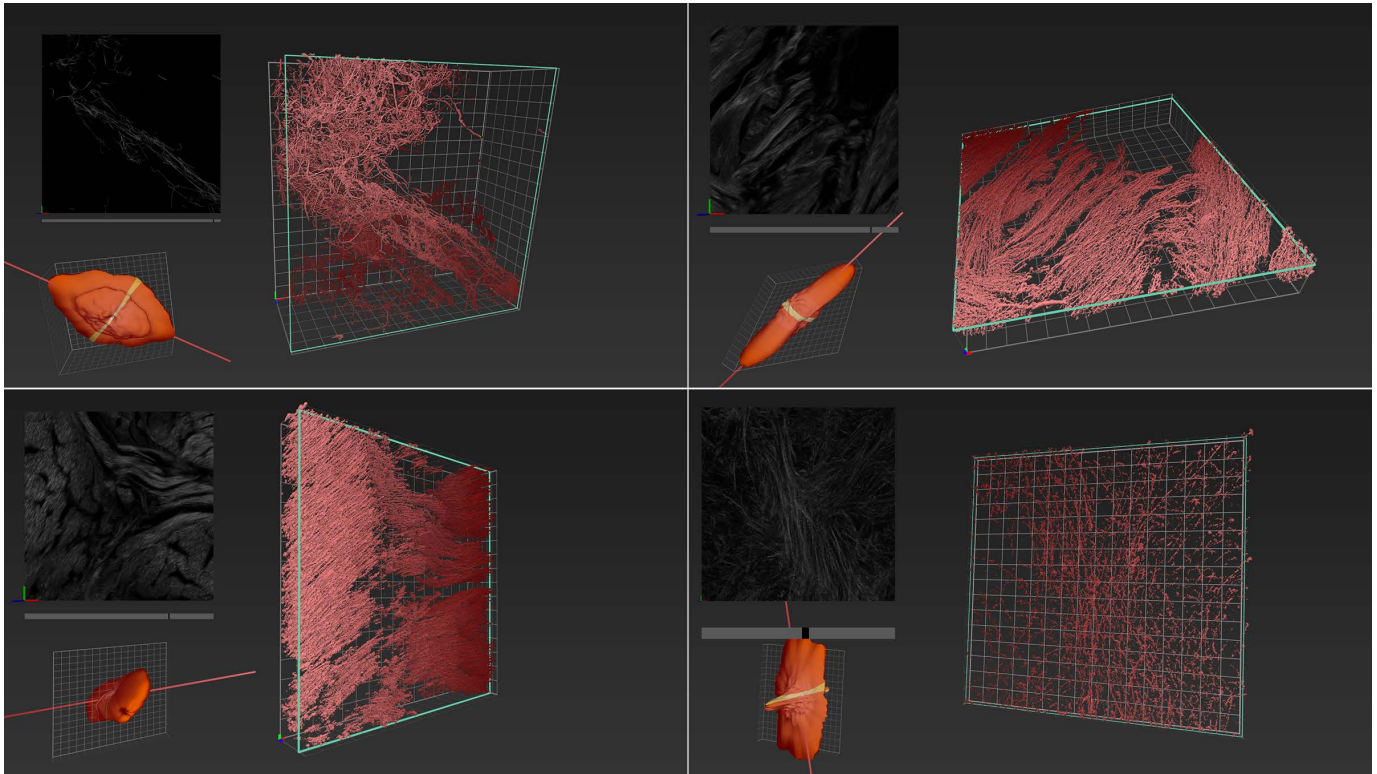


Fig. 9. A variety of tissue datasets used with the system. Upper left: a collagen scaffold. Notice how the halos, lighting, and coloring help show the varying structure of the data. Upper right: pig sclera tissue. Lower left: a cross section of pig tendon. Lower right: rat cervix tissue.

the Scattered Snakelets algorithm [23]. Snakelets are small active contours created by cutting traditional active contour snakes into shorter independent sections. These short sections are then computationally efficient to compare against the current prop orientation.

Conceptually, the algorithm works by iteratively moving snakelet endpoints towards the local maximum in the gradient of a voxel distance map. This approach is similar to the work of Prohaska et al. [20] which also uses a distance map gradient to find centerlines. Because the centerlines of fibers are furthest from the background, they contain the highest distance map values, and the snakelets gradually converge on the centerline. After convergence, adjacent and overlapping snakelets are merged by averaging their endpoints.

As opposed to the original algorithm that was designed to extract centerlines from large tubular structures like the small intestine, our extensions make it possible to apply the snakelet technique to dense biological tissues. Specifically, we use adaptive thresholding to create the Euclidean distance map, which is critical in our applications since variations in fiber density cause changes in the background intensity throughout the volume. A spherical radius is also used, rather than an orthogonal plane as in the original algorithm, to find adjacent snakelets for merging. This helps reconnect gaps caused by segmentation errors. Finally, performance is improved by removing the active contour aspect of the original algorithm, keeping the snakelets as straight segments. Given the small length and large number of overlapping snakelets, users perceive the fibers as smooth curving paths.

The algorithm contains four steps:

1. Calculate a Euclidean distance map for the fiber voxels using adaptive thresholding:
 - Set the threshold to the maximum of the average value of the voxels within a six-voxel radius and a minimum value set empirically for each dataset.
2. Using a regular grid over the voxels (we use a grid size of 2 voxels), initialize three orthogonal snakelets at each grid point

Table 1. Runtime Results for Identifying Fiber Structures.

Dataset	Dimensions	Initialized Snakelets	Final Snakelets	Runtime (sec)
Collagen Scaffold	$2048 \times 2048 \times 165$	287,905	280,860	110.81
Pig Tendon	$1024 \times 1024 \times 138$	4,282,503	538,396	868.64
Pig Sclera	$1024 \times 1024 \times 161$	3,771,408	298,780	784.45
Pig Cornea	$1536 \times 4608 \times 16$	2,764,542	395,230	464.55
Rat Cervix	$512 \times 512 \times 27$	298,995	40,219	27.77

with a positive distance value (i.e., each grid point that contains some part of a fiber):

- The initial length for these snakelets should be equal to the average fiber diameter within the dataset in order to avoid snakelets that lie within two different fibers (we use a length of 5 voxels).
3. Refine snakelet endpoints towards the fiber center:
 - For each snakelet, iteratively move each endpoint towards the local maximum in the gradient of the distance map.
 - Repeat until the endpoints move less than half of a voxel.
 4. Remove snakelets that cross fibers and merge adjacent snakelets.

7 RESULTS

Figure 9 shows screenshots from applying the visualization strategy to four different fiber datasets. Since these images capture just a point in time, the best indication of the results achieved with the system can be viewed in the accompanying video, which documents the way that users typically interact with the system.

Run times for the snakelet algorithm are reported in Table 1. The scattered-snakelet approach can be parallelized. Our implementation is parallelized using OpenMP on a machine with dual 2.5 GHz Intel Xeon E5-2640 with 6 cores each.

8 USER FEEDBACK AND EVALUATION

Our research team includes domain-science collaborators who study biophotonics and have helped us design and evaluate the system using their datasets. One is a graduate student in mechanical science and engineering. The other directs a biophotonics, optical physics, and engineering research lab at a top academic research institution. Both make regular use of computational analysis tools such as Matlab, but before starting this project neither had ever visualized their data in 3D. Both had used the Microsoft Kinect a few times previously to play games but never in a scientific context.

Our design process spanned more than one year and included more than 10 iterative design and critique sessions, with updated hardware and software tools delivered to our collaborators' site several times. They currently use the system one to two times a week to explore their data, and they estimate that they have spent over 80 hours to date using the system.

In addition to this ongoing process, we have also conducted two hour-long structured sessions to provide a more summative evaluation of the system. Since our team is separated by distance, these sessions were held over videoconferencing. Identical hardware and software systems were developed and installed at each site. Thus, one of the first bits of evaluative feedback is that the interface and algorithms presented here are robust to working in different labs, in different lighting conditions, and with different users controlling the interface.

Before the summative evaluation sessions, the domain scientists had only used prototypes of the system. At the beginning of each session, the participants were given a reminder tutorial of the visualization and interaction techniques. They were then able to use the system themselves to explore the data shown in Figure 9. While interacting with the data, they were encouraged to talk-out-loud and ask questions. Occasionally reminders about how to perform specific gestures were given. Afterwards, a semi-structured interview was conducted with notes recorded by the evaluators.

We structure the user feedback and evaluation along three axes: (1) fit within normal workspace activities, (2) utility of specific visualization features, and (3) impact within the application domain.

8.1 Axis 1: Fit within Normal Workspace Activities

High-level feedback on the system supports our goal of developing an interactive, desktop-based 3D visualization platform using consumer-level hardware. Our collaborators described new research in their field and how it might intersect with this style of data visualization. The ability to bring 3D visualization into the same physical space as the bioimaging hardware could create a paradigm shift in how bioimaging tools are used. One scientist commented, "From my perspective this is a big thing because you have brought it [the 3D visualization] to someone's desk ... it is incredible that you can buy this".

Currently, the scientists analyze their data by viewing static 2D slices or volume renderings. They use Matlab to calculate local fiber orientations, but for visualization, the information is decoupled into two separate 2D histograms of θ and ϕ spherical coordinates. In contrast to this approach, the collaborators readily confirmed that the ability to query the data and interactively display fibers at particular 3D orientations using the prop-based interface was immediately useful. They reported that they view the system as being most helpful during initial stages of data analysis, which involves exploring their data to gain insights and identify specific hypotheses that could then be quantified after visualization using more traditional quantitative tools.

8.2 Axis 2: Utility of Specific Visualization Features

Users appreciated the ability to specify a 3D vector via a physical prop. One scientist commented "Taking the wand and pointing it in 3D space is as intuitive as it can get".

Some of the more complex interactions such as scalar adjustment were not as self-revealing. One scientist said, "This is so far from what we are used to today. This is a whole new world of interaction..." His comment is representative of the excitement we observed throughout the tool development, but we also understand that the interface and visualization require scientists to adopt a new way of thinking about

their data analysis. This will take time to integrate into their current workflows. The same scientist continued, "How much ... do I have to think about what I'm doing to get a certain effect vs. can I draw upon my knowledge of how I use computers [today]".

The need to learn and remember gestures is one limitation of many gesture-based user interfaces, and it would be interesting to see if techniques that facilitate learning 2D gesture sets, such as the Gesture Bar [1], might also be applicable to prop-based 3D interfaces. Additional discussion on this theme revolved around how the paper prop is extremely simple, but does four different things. Users stated that they might actually prefer to move some of this functionality out of the prop-based interface and into more traditional interfaces, such as menu systems with which they are already familiar. We are interested in studying further whether this preference might change after more use of the system or whether a hybrid interface combining props with more traditional graphical interfaces might be more effective for some users. If so, the questions of how to facilitate seamless transitions between the different styles of interfaces and how to most appropriately assign tasks to each interface would be interesting to study.

The 3D histogram for visualizing the fiber orientation distribution was evaluated as both useful and an interesting contrast to current methods. One user mentioned that he needed to learn how to interpret it relative to the 2D histograms used in current workflows. Again, there was an opinion that the interactive 3D visualization might be the best place to discover a 3D trend, but then more traditional Matlab-based 2D tools might be the right place to perform a quantitative analysis. Moving in the direction of including additional quantitative outputs in the visualization, one specific feature that was requested is to quantify the relative amount of fibers oriented in the direction indicated by the user rather than just showing the visual distribution.

The ability to reorient the 3D volume and the 3D histogram was evaluated as critical and effective. The domain scientists mentioned that this would let them explore where the interesting data features are and then start to develop the story that explains them. This led to a series of insights about useful extensions of the current interface. Extending the current approach of selecting subsets of the fibers based upon orientation, it would be useful to also be able to narrow the scope of the visualization to a smaller portion of the volume or zoom in to look at a feature in more detail.

8.3 Axis 3: Impact within the Application Domain

The domain scientists were enthusiastic about how this system could change their workflow, and as a result, the science they can do today. It is clear that trends can be observed in this interactive 3D visualization that are not easily discovered using current tools. The ability to see the fibers in a true 3D spatial context is regarded as a major change in this discipline. It is not yet clear exactly what impact this can have, but there is great excitement about the potential, and if visualization systems in this style can continue to advance toward the coupled real-time imaging and visualization that we describe as motivation in the introduction to the paper, this would be transformative in the field.

9 EXTENSIONS FOR ADDITIONAL APPLICATIONS

In terms of the broader applicability of the approach, we see the potential for the system to be generally applied to exploring vector-field datasets, such as fluid flow, weather data, transportation patterns, or marker trajectories from motion capture. In these applications, the prop orientation could be used to filter the display of vectors or segments of trajectories.

In addition, feedback from the domain scientists indicates strong potential for the system to be broadly applied to bioimaging datasets in addition to those shown in Figure 9. Nerve fibers were suggested as a specific useful next application area since there is even more variation in their spatial arrangements. Although many bioimaging datasets — including the ones referenced in this paper — contain straight fibers, curving fibers are also common (e.g., neural fibers, blood vessels). One way to extend the interface to work with these datasets could be to utilize flexible props, perhaps in the style of recent flexible displays [15].

10 LIMITATIONS

One limitation of the current implementation is that the roll gesture requires users to roll the prop slowly enough that motion blur from the limited resolution of the low-cost depth camera does not cause mis-readings of the Gray code pattern. In general, we found that the tracking required to implement this interface today is right at the boundary of what can be successfully accomplished using today's commodity depth-sensing camera hardware. Although other low-cost interaction hardware, such as the wand-based Playstation Move, might remove this limitation, tracking based on depth-sensing allows for a wider range of interaction techniques. For instance, it enables the software to detect whether the prop is held with one or two hands. It is not unreasonable to expect that computer webcams will be replaced with depth-cameras in the next 5–10 years, and as the camera resolution increases, we think that the paper prop could be replaced with a pen or pencil that are readily available on a scientist's desk. Looking towards this future, we are motivated to explore "lightweight" props rather than something like the "3D wand" used for the Playstation Move, which contains a fair amount of active electronics inside.

Our current implementation with just a single depth camera does lose tracking of the prop when it is held at an orientation that points directly into the display. In practice, this is not a problem for our particular application and datasets. Since the volumes are all smaller along one dimension, we orient this dimension into the screen, and if there is a need to specify an orientation that runs directly into the screen, the user simply reorients the volume. This limitation should also be able to be addressed with additional hardware (e.g., 2 cameras).

Finally, like most in-air gesture-based interfaces, there is some potential for user fatigue when using the system; however, the system is designed to track props at close range to the camera, enabling the user to rest his or her arms on the desk, which minimizes fatigue.

11 CONCLUSIONS AND FUTURE WORK

This work demonstrates the extent to which commodity visualization technologies, such as a depth sensing camera and low-cost 3D display, can now be combined to produce an interactive spatial interface with passive haptic feedback within a scientist's office workspace. In the future, we are keen to extend this interface to support additional operations, such as controlling the SHG microscope. This could enable real-time streaming of data from the microscope to the visualization system, which would have a great positive impact on the imaging pipeline. More broadly, we are also excited to continue to advance the concept of "paper-based interactive visualization". We believe that there are many ways that regular materials (e.g., pens, paper) that we use naturally in our everyday work and discussions with collaborators might be employed to support more effective methods of interacting with, annotating, and exploring computer-based data visualizations. We believe this mixing of the real and physical world can often facilitate collaboration and make our visualizations more effective, more engaging, and more meaningful.

ACKNOWLEDGMENTS

This work was supported by the National Academies Keck Futures Initiative and the National Science Foundation (IIS-1218058).

REFERENCES

- [1] A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola, Jr. Gesturebar: improving the approachability of gesture-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2269–2278. ACM, 2009.
- [2] N. Burrus. Kinect calibration. <http://burrus.name/index.php/Research/KinectCalibration>, June 2013.
- [3] P. Campagnola, M. A. Brewer, V. Ajeti, P. Keely, K. Eliceiri, M. Patankar, and K. Tilbury. SHG imaging of cancer. In *Biomedical Optics*, page BSu4B.1. Optical Society of America, 2012.
- [4] G. Casiez, N. Roussel, and D. Vogel. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530. ACM, 2012.
- [5] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, and D. F. Keefe. Interactive slice WIM: Navigating and interrogating volume datasets using a multi-surface, multi-touch VR interface. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1614–1626, 2012.
- [6] M. H. Everts, H. Bekker, J. B. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1299–1306, 2009.
- [7] J. Goble, K. Hinckley, R. Pausch, J. Snell, and N. Kassell. Two-handed spatial interface tools for neurosurgical planning. *Computer*, 28(7):20–26, 1995.
- [8] R. Held, A. Gupta, B. Curlless, and M. Agrawala. 3D puppetry: a kinect-based interface for 3D animation. In *Proceedings of the ACM symposium on User interface software and technology*, pages 423–434, 2012.
- [9] D. Herrera, J. Kannala, and J. Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 2012.
- [10] V. Interrante and C. Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In *Proceedings of the 8th conference on Visualization*, pages 421–424. IEEE Computer Society Press, 1997.
- [11] B. Jackson and D. F. Keefe. Sketching over props: Understanding and interpreting 3D sketch input relative to rapid prototype props. IUI 2011 Sketch Recognition Workshop, 2011.
- [12] B. Jackson, D. Schroeder, and D. F. Keefe. Nailing down multi-touch: Anchored above the surface interaction for 3D modeling and navigation. In *Proceedings of Graphics Interface*, pages 181–184. Canadian Information Processing Society, 2012.
- [13] D. F. Keefe and T. Isenberg. Re-imagining the interaction paradigm for scientific visualization. *IEEE Computer*, 45(5):51–57, May 2013.
- [14] T. Klein, F. Guéniat, L. Pastur, F. Vernier, and T. Isenberg. A design study of direct-touch interaction for exploratory 3D scientific visualization. *Computer Graphics Forum*, 31(3pt3):1225–1234, 2012.
- [15] J. Konieczny, C. Shimizu, G. Meyer, and D. Colucci. A handheld flexible display system. In *Proceedings of IEEE Visualization*, pages 591–597, 2005.
- [16] S. Krucinski, I. Krucinska, S. Veeravanallur, and K. Slot. Computer-assisted analysis of the extracellular matrix of connective tissue. In *Proceedings of SPIE Medical Imaging*, volume 3034, pages 950–963, 1997.
- [17] K. Kruszyński and R. Liere. Tangible props for scientific visualization: concept, requirements, application. *Virtual Reality*, 13(4):235–244, 2009.
- [18] A. Miller, B. White, E. Charbonneau, Z. Kanzler, and J. LaViola. Interactive 3D model acquisition and tracking of building block structures. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):651–659, 2012.
- [19] M. R. Mine, F. P. Brooks, Jr., and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM, 1997.
- [20] S. Prohaska and H.-C. Hege. Fast visualization of plane-like structures in voxel data. In *Proceedings of IEEE Visualization*, pages 29–36, 2002.
- [21] A. J. Sellen, G. P. Kurtenbach, and W. A. S. Buxton. The prevention of mode errors through sensory feedback. *Human-Computer Interaction*, 7(2):141–164, June 1992.
- [22] M. Sivaguru, S. Durgam, R. Ambekar, D. Luedtke, G. Fried, A. Stewart, and K. C. Toussaint. Quantitative analysis of collagen fiber organization in injured tendons using fourier transform-second harmonic generation imaging. *Optics Express*, 18(24):24983–24993, 2010.
- [23] C. Spuhler, M. Harders, and G. Székely. Fast and robust extraction of centerlines in 3D tubular structures using a scattered-snakelet approach. In *Proceedings of SPIE Medical Imaging*, volume 6144, pages 1295–1302, 2006.
- [24] J. P. Wachs, H. I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith. A gesture-based tool for sterile browsing of radiology images. *Journal of the American Medical Informatics Association*, 15(3):321–323, 2008.
- [25] A. Wenger, D. F. Keefe, S. Zhang, and D. H. Laidlaw. Interactive volume rendering of thin thread structures within multivalued scientific datasets. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):664–672, 2004.
- [26] J. Wu, B. Rajwa, D. L. Filmer, C. M. Hoffmann, B. Yuan, C.-S. Chiang, J. Sturgis, and J. P. Robinson. Analysis of orientations of collagen fibers by novel fiber-tracking software. *Microscopy and Microanalysis*, 9(6):574–580, 2003.