

# Open-source surface mesh-based ultrasound-guided spinal intervention simulator

Laura Bartha · Andras Lasso · Csaba Pinter ·  
Tamas Ungi · Zsuzsanna Keri · Gabor Fichtinger

Received: 11 January 2013 / Accepted: 17 May 2013 / Published online: 1 June 2013  
© CARS 2013

## Abstract

**Purpose** Ultrasound is prevalent in image-guided therapy as a safe, inexpensive, and widely available imaging modality. However, extensive training in interpreting ultrasound images is essential for successful procedures. An open-source ultrasound image simulator was developed to facilitate the training of ultrasound-guided spinal intervention procedures, thereby eliminating the need for an ultrasound machine from the phantom-based training environment.

**Methods** Anatomical structures and surgical tools are converted to surface meshes for data compression. Anatomical data are converted from segmented volumetric images, while the geometry of surgical tools is available as a surface mesh. The pose of the objects are either constants or coming from a pose-tracking device. Intersection points between the surface models and the ultrasound scan lines are determined with a binary space partitioning tree. The scan lines are divided into segments and filled with gray values determined by an intensity calculation accounting for material properties, reflection, and attenuation parameters defined in a configuration file. The scan lines are finally converted to a regular brightness-mode ultrasound image.

**Results** The simulator was tested in a tracked ultrasound imaging system, with a mock transducer tracked with an Ascension trakSTAR electromagnetic tracker, on a spine phantom. A mesh model of the spine was created from CT data. The simulated ultrasound images were generated at a speed of 50 frames per second, and a resolution of  $564 \times 597$  pixels, with 256 scan lines per frame, on a PC with a 3.4 GHz

processor. A human subject trial was conducted to compare the learning performance of novice trainees, with real and simulated ultrasound, in the localization of facet joints of a spine phantom. With 22 participants split into two equal groups, and each participant localizing 6 facet joints, there was no statistical difference in the performance of the two groups, indicating that simulated ultrasound could indeed replace the real ultrasound in phantom-based ultrasonography training for spinal interventions.

**Conclusions** The ultrasound simulator was implemented and integrated into the open-source Public Library for Ultrasound (PLUS) toolkit.

**Keywords** Ultrasound simulation · Surface mesh · Training · PLUS toolkit

## Introduction

Ultrasound imaging is often chosen for guiding interventions. Patients are not required to undergo any radiation, or enter into any magnetic fields, and it is also significantly less expensive than other modalities, such as computed tomography (CT) and magnetic resonance imaging (MRI). It is also portable, allowing for the imaging equipment to come to the patient, be they in the operating room, or any other room in the hospital. However, extensive training is necessary to make use of its full potential. Ultrasound is notorious for providing images with features that are only discernible with experience. This is problematic; in that, an ultrasound machine would be needed for every trainee in the process of acquiring the skills of ultrasound diagnosis. Simulation could be the solution to this problem, potentially relieving some of the financial burden associated with obtaining numerous ultrasound machines.

L. Bartha (✉) · A. Lasso · C. Pinter · T. Ungi · Z. Keri · G. Fichtinger  
Laboratory for Percutaneous Surgery, School of Computing,  
Queen's University, Kingston, ON, Canada  
e-mail: bartha@cs.queensu.ca

G. Fichtinger  
e-mail: gabor@cs.queensu.ca

Ultrasound simulation has been studied in the literature since the 1970s. The first approaches sought to model the propagation of the ultrasound wave based on physical models. Of such generative methods, Jensen's FIELD II simulator [6, 7] is well known. Developed in the 1990's, it remains an influential piece of work and free to use, but the source code is not openly available. Almost all of the contending simulators use a pre-existing three-dimensional image volume, acquired from a real ultrasound, CT, or MRI scan. This approach is inherently very different from the Field II simulator. CT-based simulators proposed by Hostettler et al. [5], Magee et al. [13], and Wein et al. [21] have contributed greatly to the field of ultrasound simulation, introducing novel methods to generate ultrasound images. Wein et al. [21] employ a transfer function to infer acoustic impedance values with the use of the Hounsfield units in the CT. However, very high-resolution CT volumes are needed if fine details are to be preserved, and two areas that appear the same on a CT could have very different acoustic properties. Recently, the graphics processing unit (GPU) has facilitated the use of more complex algorithms [2, 8, 9, 20] in the processing of the three-dimensional image volumes, allowing for the use of more complicated methods while maintaining image generation at a speed comparable to clinical practice. The GPU has even been used for more updated generative approaches to simulation, such as Karamalis et al. in 2010, who modeled ultrasound wave propagation and interaction with different media using the finite-difference scheme to solve the Westervelt equation [9]. Gjerard et al. [2] also employed the GPU, using it to numerically convolve the point-spread function with the tissue model, based on the fast Fourier transform.

There has been significant amount of work on simulating ultrasound from previously acquired medical images. The focus of more current research has shifted from generating accurate tissue simulation, with proper speckle and other physical parameters, to implementing only a slicing of the original volume with minimal processing. These works were not the first to use reslicing, but they did focus on other interesting aspects of simulation. The goal of Zhu et al. [22] was to realistically simulate a needle in the ultrasound image, which can be challenging, as the appearance of a needle changes depending on the angle it is imaged from. The work of Goksel and Salcudean centered on simulating deformation in previously recorded ultrasound images of tissue phantoms [3]. Skehan [17] built upon a previously existing system, which used existing 3D volumes to extract a 2D slice. The works of the different groups mentioned above are heavily based on pre-acquired ultrasound volumes, which require access to an ultrasound machine. Many do not account for the angle-dependent variations throughout the image.

Simulating ultrasound specifically from CT volumes has become quite different from using other imaging modalities since Wein et al. made use of the partial relationship between

the intensity of the CT image, expressed in Hounsfield units, and the acoustic impedance in ultrasound. The Hounsfield units are approximately proportional to the tissue density, which is in turn approximately proportional to the acoustic impedance [21]. This provides a means to simulate large-scale ultrasound reflection at tissue boundaries, as well as shadowing effects at bone interfaces. However, the unique backscattering pattern of each tissue type visible in ultrasound, that is, the echogenicity, cannot be calculated from this relationship, as there are many other contributing factors to consider, including imaging depth. In their attempt to remedy this, Wein et al. [21] developed an intensity mapping on a narrow soft tissue range based on a number of correspondences between CT intensities and tissue echogenicity. This method has been highly influential and many later works that simulate ultrasound from CT use Wein's method. Such an example is the simulation developed by Gill et al. [1]. They decided on modeling the ultrasound passing through the CT volume as a ray, calculating the reflection and transmission using the equations developed by Wein et al. [21] at each voxel. The drawback with Wein's method is that the proportional Hounsfield unit to acoustic impedance relationship does not apply to all tissue, and the method cannot be easily expanded to other imaging modalities. In general, CT-based simulation methods are very limited to the cases that they can train, as each new scenario would require a new CT.

Ultrasound simulation, from CT volumes in more recent papers, often takes advantage of the graphics processing unit (GPU). Nevertheless, Wein's method has remained highly influential, such as in works by Kutter et al. [10] and Karamalis [8]. Vidal et al. [20], however, strays from Wein and makes full use of the image processing capabilities inherent in the GPU. Kutter et al. [10] still use a ray-based model and draw upon the Hounsfield unit—acoustic impedance proportionality. Ray sampling is also employed by Karamalis et al. [8]. Very much like Kutter, a ray is calculated that propagates through the CT volume beginning at each transducer element. The work of Vidal et al. used a very different methodology that incorporates visualizing needle insertions in the ultrasound. Once they obtained a 2D oblique image from parallel CT slices, they calculated the needle position and replaced those pixels in the oblique image with bright pixels simulating the reflection produced by the metallic needle [20]. The disadvantage associated with methods employing the GPU is that the resulting code tends to become increasingly platform dependent, and its performance is more difficult to guarantee on different computers.

The work presented in this paper focuses on ultrasound simulation for spinal intervention training scenarios using phantoms, with the goal to eliminate expensive ultrasound machines from ultrasound training scenarios that involve the use of phantoms. In typical phantoms, the soft tissues usually have little or no speckle and, especially in musculoskeletal

phantoms, bony structures dominate the view. The proposed ultrasound simulation method exploits this observation.

## Methods

We introduce a surface-based geometrical model for the simulation. This model allows production of simulated images that clearly show the shape of anatomical structures. As each structure is represented by a separate mesh, the complexity of the displayed image—and thus the difficulty of the training task—can be controlled flexibly by showing only the relevant anatomical structures, while hiding others. Moreover, the surface model can be easily altered computationally in any design program allowing for the study of an infinite variety of pathological cases or to show time-varying models. Surface meshes cannot describe the internal texture of the structures, but it is argued that this noise texture may not be always necessary or even prove to be a distraction during the learning process in certain cases, especially if the objective is the recognition of the shapes of anatomical structures. The surface mesh-based simulator is implemented entirely at the level of application software in a system-independent manner, and hence, it is particularly amenable to multi-platform open-source implementation, and thereby widespread use.

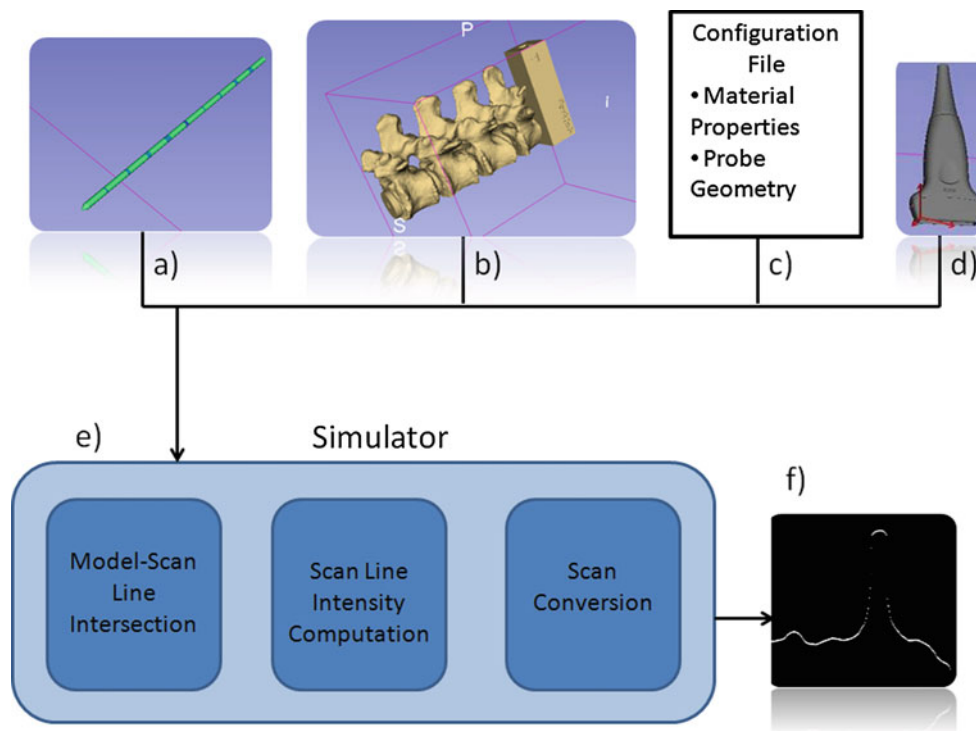
From conception, the simulator was designed to take advantage of the Visualization Toolkit (VTK) [15, 16] and to be integrated into the Public Library for Ultrasound (PLUS) software toolkit [11, 12]. This offered easy access to a wide range of functions for image, surface mesh, and pose information acquisition and processing. The input to the simulator is a surface mesh model and a linear transformation representing the spatial relationship between the surface mesh and the simulated ultrasound image. This allows the model to be transformed to the image coordinate system, and the intersecting region to be identified and displayed as a constant intensity binary image. The linear transformation can either be specified as a constant, in a configuration file, or come from a tracker. The surface mesh, if anatomical, can be generated easily from any segmented volumetric image, such as a CT or MRI scan. However, the segmentation itself can be a time-consuming process. In our case, the surface mesh was of a spine originally from a CT scan. Surface meshes of surgical tools are readily available with such medical image software, and surface meshes in general are a compact form of data representation when compared to volumetric models. The pose, that is, the linear transformation mentioned above, is obtained either from a live tracker or from a configuration file. Transducer parameters, such as radius, imaging depth, number of elements, and shape (curvilinear versus linear), are also defined in the configuration file read in by the simulator, as are the material properties associated with each model. The workflow of the simulator is described in Fig. 1.

The presently used functional and practical ultrasound simulator emerged through three evolutionary stages. All three methods used the same inputs: a surface mesh, configuration file, and a linear transform representing pose information. The output produced was also the same: the simulated ultrasound image.

The heart of our initial approach was generating a preliminary binary image, which involved the creation of a new VTK filter, comprised of existing filters. The new filter performed a series of preprocessing steps to prepare it for the `vtkPolyDataToImageStencil` filter, resulting in a stencil in the shape of a cross section of the model, at the position of the imaging plane (Fig. 2a). Algorithmically and geometrically, this method worked correctly, but it was far too slow, needing 40 min to produce about 80 frames. This method lacked the requisite speed, due to the fundamental problem of transforming the dense model each time the position of the image plane, which corresponds to the position of the ultrasound probe, changed. The straightforward fix would be to transform the image plane to the model coordinate system, but this is not possible in the VTK framework, as images have no orientation.

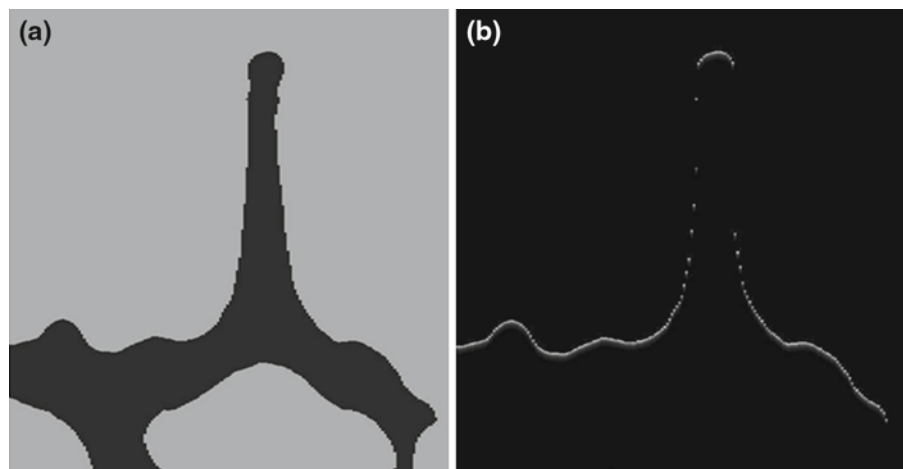
In the second method, the “filter of filters” construction was retained, and a new version of the `vtkPolyDataToImageStencil` filter was developed. This filter was able to support images in any orientation with the use of `vtkPlanes` to store the relevant transformations. To further increase the speed, a binary space partition tree (BSP) was used to find all intersecting positions of the image plane and the model of the bone. This approach allowed for processing 4 frames per second, which was a huge improvement compared to our first method, but it was still inadequate for use in a real-time clinical training system.

In the third and presently used final method, we abandoned the previously used “filter of filters” construction for ray tracing altogether. The input still consists of the surface mesh and the transformation. The simulated ultrasound image is generated in two steps: first, a scan line computation and then a scan conversion. Scan line positions are determined in the image coordinate system using the desired size of the simulated ultrasound image, a parameter that is specified in a configuration file. These positions are then transformed to the model coordinate system, and a BSP tree (`vtkModifiedBSPTree`) is used to determine the points intersecting with the model located along the scan line. The intersection points divide the scan line into segments, which are then filled with a gray value, the result of an intensity calculation based on the material properties also defined in the configuration file. The process of dividing a scan line into such segments is depicted in Fig. 3. The scan lines are finally converted to a regular brightness-mode ultrasound image using the RF processing module of PLUS. The RF processing module interprets the information in the scan lines and applies



**Fig. 1** Workflow of the simulator; *a* shows an example of a surface model, in the form of a needle, *b* shows another surface model in the form of a spine, *c* represents the configuration file used by the simulator, *d* exemplifies the linear transformation representing the probe position

in relation to the surface mesh, *e* represents the simulator divided into three steps, and *f* shows a sample-simulated image as the result of the workflow

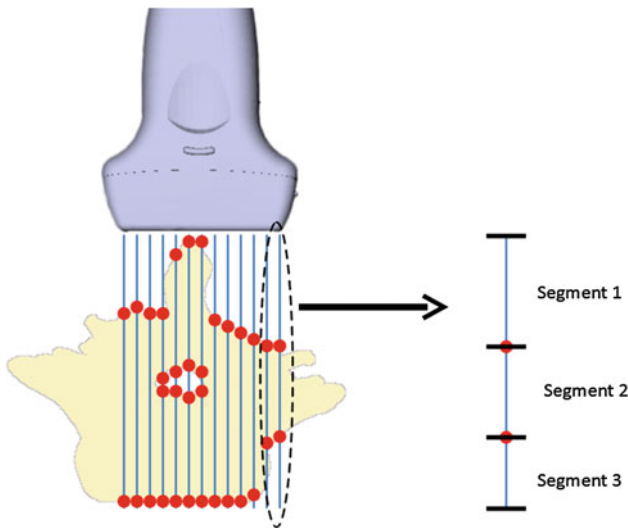


**Fig. 2** **a** Cross-sectional image of a vertebra resulting after the first stage of simulator development. **b** Simulated ultrasound image of a vertebra generated using the scanline method

the appropriate spacing to formulate an ultrasound image. The scan line method allows for relatively straightforward computation of the gray values of the generated image, facilitating more complicated intensity calculations based on the physical properties of ultrasound. Strong reflections from the bone and background surface, as well as fast absorption in the bone, are simulated. Once an intersection point is detected, a

short series of bright pixels, rapidly decreasing in intensity, are written to that segment of the scan line, and the rest of the scan line is filled with black. A sample of such a simulation is displayed in Fig. 2b).

The behavior of the ultrasound simulator is governed by the configuration file used in the PLUS toolkit, in which the key parameters and working characteristics of the ultrasound



**Fig. 3** The *large dots* represent the intersection points between the image plane and the surface model. *Straight lines* are the scanlines divided into different segments by the intersection points that fall along the scan line

scanner are specified. A typical example of the configuration file is shown in Fig. 4. New parameters were added to the configuration file to describe the specific tissue properties, including the attenuation coefficients, density, and sound velocity. These new parameters, along with the distance travelled in the medium, are used during scan line computation to calculate the intensity of each pixel. According to Hedrick et al. [4], total intensity loss at each pixel is comprised of loss due to attenuation, which is defined as the combined effect of scattering and absorption, and loss due to reflection

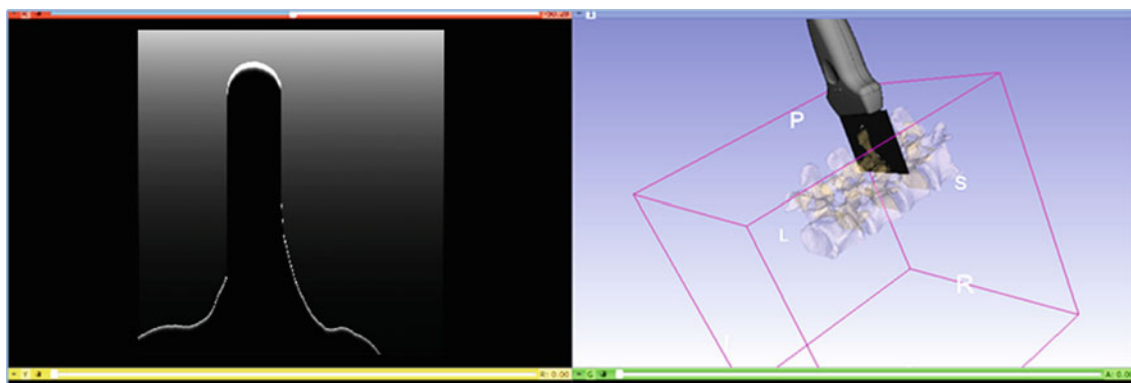
at tissue boundaries. The loss due to attenuation is calculated using the attenuation coefficient parameter found in the configuration file, and reflection is taken into consideration at tissue boundaries using the sound velocity and density parameters, also in the configuration file. It is of note that the configuration file accommodates multiple simulated models, the calculation of which has not yet been implemented in the present prototype.

The completed single model simulator is presently fully functional in a 3D Slicer environment [15]. 3D Slicer is a free, open-source, highly customizable, and extensible software package for the visualization and analysis of medical images. Anatomical data are acquired from segmentations of CT or MRI data converted into surface meshes. Other objects, such as surgical tools, are also represented using surface meshes, allowing for the preservation of model details. The pose of the objects is represented using linear transformations, which could be specified either as constants, in a configuration file, or coming from a tracker. Transducer parameters, such as radius, imaging depth, number of elements, and shape (curvilinear or linear) are also defined, along with the material properties associated with each model. The PlusServer application of the PLUS toolkit connects to a tracking and an ultrasound imaging device, and continuously sends the acquired data for visualization through the OpenIGTLink [18] protocol to 3D Slicer. PlusServer can use a wide range of optical and electromagnetic tracker devices and various ultrasound imaging devices, including our simulator or real ultrasound scanners. This particular experimental setup was tested and described below in the results section. The ultrasound simulator was implemented in C++, as part of the open-source PLUS toolkit [11, 12] available at <http://www.plustoolkit.org>.

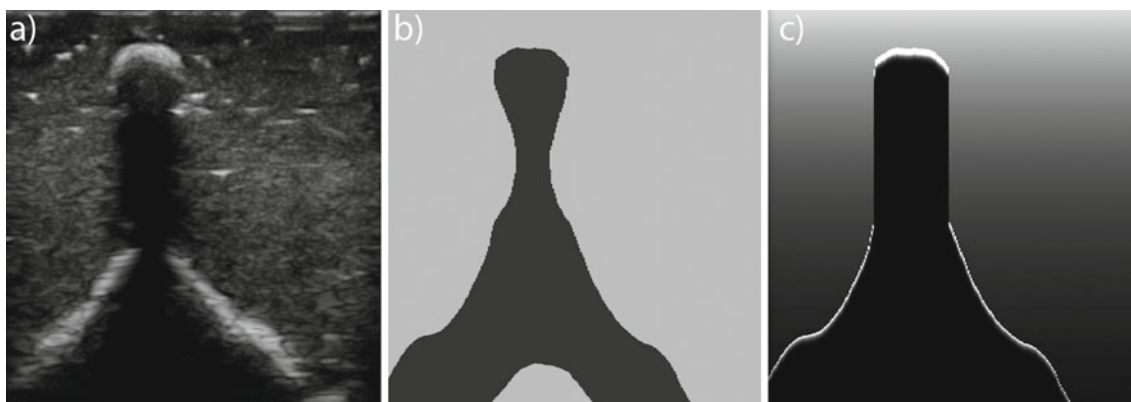
```
<vtkUsSimulatorAlgo
  BackgroundValue="155" ImageCoordinateFrame="Image"
  ReferenceCoordinateFrame="Phantom" NumberOfScanlines="256"
  NumberOfSamplesPerScanline="1000" FrequencyMHz="2.5"
  MaxIntensityWattsPerCm2="5000">
  <SceneDescription BackgroundMediumId="background">
    <SpatialModel Id="background"
      Name="Soft tissue"
      DensityKgPerM3="1080" SoundVelocityMPerSec="1580"
      AttenuationCoefficientNpPerCm="0.38" ScatterCoefficientNpPerCm="0.2"
      SpecularReflectionRatio="0.0" AbsorptionCoefficientNpPerCm="0.2"/>
    <SpatialModel
      Name="Spine" ModelFileName="Spine.stl"
      ModelFileToSpatialModelTransform="1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1"
      DensityKgPerM3="1912" SoundVelocityMPerSec="4080"
      AttenuationCoefficientNpPerCm="2.3" ScatterCoefficientNpPerCm="1.5"
      SpecularReflectionRatio="0.5" AbsorptionCoefficientNpPerCm="4.2"/>
    <SpatialModel
      Name="Needle" ModelFileName="Needle.stl"
      ModelFileToSpatialModelTransform="1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1"
      DensityKgPerM3="7860" SoundVelocityMPerSec="5130"
      AttenuationCoefficientNpPerCm="2.3" ScatterCoefficientNpPerCm="1.5"
      SpecularReflectionRatio="0.8" AbsorptionCoefficientNpPerCm="3.0"/>
  </SceneDescription>
```

**Fig. 4** Simulator section excerpt from a typical configuration file





**Fig. 5** Simulator in the 3D Slicer workspace



**Fig. 6** Images of a spinous process in the spine phantom: **a** acquired with an ultrasound scanner, **b** corresponding simulated cross section, and **c** final simulated bone image

## Experiments and results

### Visual performance tests

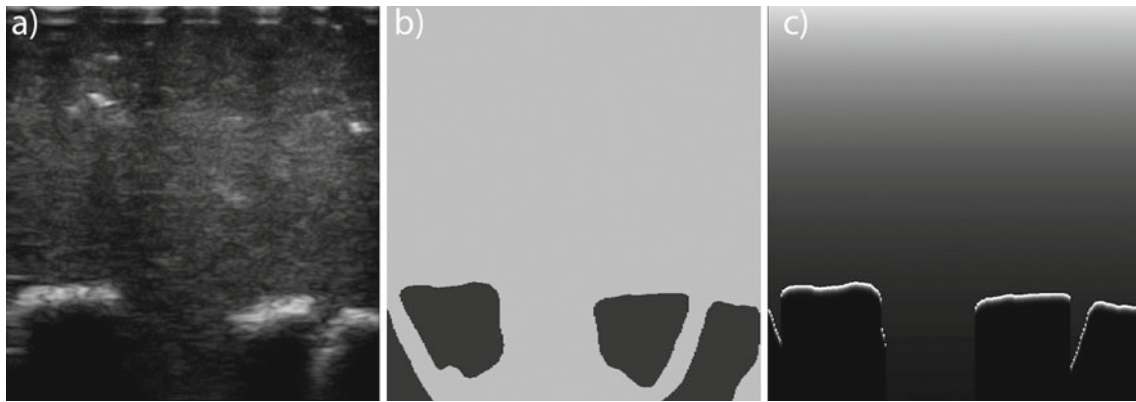
The simulator was used in the Perk Tutor ultrasound-guided spinal injection training system [19]. The anatomical object (spine), tools (ultrasound transducer, needle), and simulated ultrasound image were visualized in 3D Slicer (Fig. 5), a key component of the Perk Tutor system. The Perk Tutor system is completed by an electromagnetic tracker, trakSTAR by Ascension, which was connected to the computer to provide position information, and a spine phantom. The purpose of this setup was to check the functionality of the simulator. Under usual circumstances, the Perk Tutor does require an ultrasound machine to image the phantom. The spine mesh used to simulate the ultrasound images and displayed in 3D Slicer consisted of over 97,000 points in 195,000 cells. The ultrasound images were generated at a speed of 50 frames per second, and a resolution of  $564 \times 597$  pixels, with 256 scanlines per frame, on a PC with a 3.4GHz processor. Figures 6 and 7 depict an ultrasound image of a spine phantom in (a) for easy comparison to the images simulated at the same position using the surface model. The (b) images

are the cross sections created after the first stage of development, and the (c) images are the final images resulting from simulating bone.

As shown in Fig. 5, the simulator was successfully integrated in the 3D Slicer environment. The PLUS Server was used to send the position information of a fake transducer to the simulator and then to send the simulated image to 3D Slicer, where it was displayed using the Volume Reslice Driver module. The simulated ultrasound image correctly followed the motion of the tracked fake ultrasound probe without noticeable delay, as this was visually ascertained by multiple independent observers in numerous series of qualitative visual testing.

### Human subject study

To test the effectiveness of the developed simulator as a training tool, a human subject study was undertaken with the Perk Tutor [18]. Previously, a study by Moulton et al. [14] found that students who had received training with the Perk Tutor and real ultrasound imaging performed significantly better in facet joint identification and subsequent injection than students who did not receive Perk Tutor training. In our study, we



**Fig. 7** Images of a facet joint in the spine phantom: **a** acquired with an ultrasound scanner, **b** corresponding simulated cross section, and **c** final simulated bone image

aimed to prove that students who were trained with the Perk Tutor do similarly well with simulated and real ultrasound. After initial training with either simulated ultrasound or real ultrasound, subjects were asked to locate the facet joint on an opaque spine phantom using ultrasound. The spine phantom included a replica of a human spine under multiple layers of artificial tissue and skin, as described in [19]. During the final localization of the facet joints, the subjects were not permitted any form of augmented reality guidance. We recruited 22 participants and divided them randomly into two equal groups of 11 subjects. The first group received training with real ultrasound, and the second group received training with simulated ultrasound. The training session entailed the trainee moving an ultrasound transducer connected to a Sonix Touch (Ultrasonix Medical Corp., Richmond, Canada) ultrasound machine across the aforementioned phantom that contained a plastic printout of a spine under artificial tissue. In addition, a 3D visualization of the spine and either simulated or real ultrasound images were displayed on a computer screen. The ultrasound images were displayed in the same 3D view as the spine as well as in a separate 2D dimensional window. In the 3D view, the position of the ultrasound relative to the 3D spine was representative of the physical location of the ultrasound transducer and phantom. This was achieved by electromagnetically tracking the probe and a reference point on the phantom, and through registering selected points on the phantom with the corresponding points on the virtual spine. After the training session, the trainee was asked to locate 6 facet joints in the phantom using real ultrasound, regardless of which training group they were in. The trainee was instructed to indicate when the facet joint was found, at which time the study coordinator took a snapshot of the ultrasound image the trainee believed to contain a facet joint. The trainee then placed a marker on the snapshot at the location where he/she believed the facet joint was located. The 6 markers were saved and later assessed by an independent physician who was not involved in the trial. Each trainee was

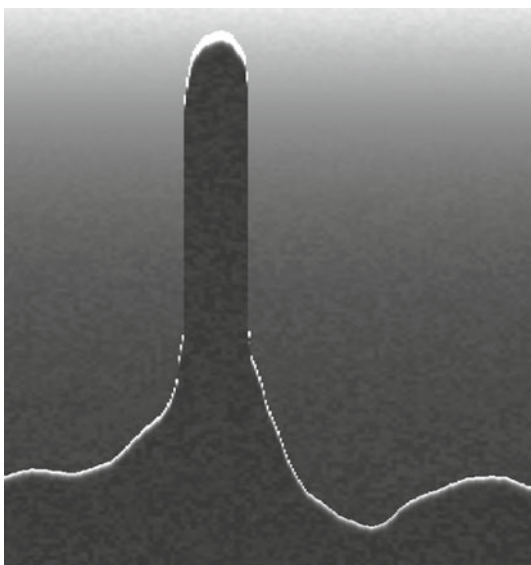
assigned a score on a scale of 0 to 6, indicating the number of facet joints identified correctly.

The simulated ultrasound trained group (called simulated group) was found to have a mean score of 4/6 facet joints correctly localized, whereas the real ultrasound trained group (called real group) had a mean score of 5/6. However, the median for the simulated group was 6 facet joints, and the median for the real group was 5. As this data are nonparametric, a Mann–Whitney *U* test was performed to compare the trainee performance, resulting in a *p* value of 0.79. This *p* value is above the conventional threshold value of 0.05, indicating that at this sample size, there is no statistical difference in the scores between the two groups. Despite the limited sample size, the mean and median scores for the two groups only differ by 1 on the scale of 6. This is a promising result in demonstrating the equivalent efficacy of simulated and real ultrasound as an imaging training tool in spinal interventions.

## Discussion

As the simulator was developed as part of the PLUS toolkit, it is immediately accessible to research groups worldwide. This also allows it to easily be used in 3D slicer, which entails a quick transition to clinical testing scenarios. It is part of a development framework that is aimed at putting medical imaging tools in the hands of physicians quickly.

An important potential use of the ultrasound simulator is producing test data for various ultrasound image processing and analysis algorithms, such as volume reconstruction and spatial and temporal calibration. The main advantage of using simulated images is the availability of ground-truth information and the ability to completely control all imaging and pose-tracking parameters. The ultrasound simulator can provide images of geometrical objects, such as ultrasound images of a surgical tool (e.g. needle, catheter, tissue



**Fig. 8** Image of a spine phantom with simulated speckle using Perlin noise

ablator, etc.), as they are being inserted into the body under ultrasound image guidance. This will allow for quantitative ground-truth testing of ultrasound image manipulation features, like the ones implemented in the PLUS toolkit. Using the simulator to generate test data is advantageous, because the number of images generated, as well as their location in relation to the surface mesh to be imaged, can be determined through a configuration file. The images can be generated in a purely virtual manner, without the use of a real ultrasound scanner, tracker, and surgical tools, quickly and efficiently.

Reproduction of various background noise textures and ultrasound reverberations may be useful for certain training scenarios. However, it is arguable that background noise may not be necessary and in certain cases may even prove to be a distraction from recognizing the shapes of the spine forms in the ultrasound image. If a trainee is given a highly realistic ultrasound image to train from, the speckle itself could prove to be distracting from the shape of spinal landmarks that the trainee must learn to recognize. The shape could be difficult to detect especially if the novice trainee does not know what he or she should be looking for. In the phantom spinal region, there is relatively little simulated soft tissue on the bony structures that tend to dominate the image. Moreover, imaging the soft tissue anatomy during phantom-based training is generally irrelevant to spinal interventions, where the anatomical target is defined by bony landmarks. Nevertheless, adding tissue-specific noise patterns to the simulated images is achievable with a straightforward extension of the proposed method, by defining a procedural texture for each structure and superimposing this texture on the scanline pixels. As an example, we added Perlin noise texture to the

generated image as shown in Fig. 8. This flexible framework allows for generating fully customizable and tissue-specific noise patterns, so that we can teach the trainee on increasingly more realistic noise and speckle patterns. The effect of these noise patterns on trainee performance remains to be a topic for future study.

To conclude, our surface-based ultrasound simulator questions whether an all-out soft tissue simulation is truly necessary for an effective musculoskeletal sonography training tool. Many of the previous works have gone to great lengths to simulate background tissue and speckle, but in spinal phantom training scenarios where soft tissue is barely visible next to bony features, this may not be needed for training. Indeed, a human subject study with the Perk Tutor showed no statistical difference between learning performance with simulated and real ultrasound in localizing facet joints in a spinal intervention training phantom. To the extreme, it may as well be desirable to suppress speckle altogether in the early stages of musculoskeletal sonography training of novice users, when soft tissue speckle may even prove to be a distraction for a trainee in learning to identify and localize skeletal landmarks and shapes. Further studies with larger sample sizes are needed to investigate this hypothesis.

**Acknowledgments** This work was supported through the Applied Cancer Research Unit program of Cancer Care Ontario with funds provided by the Ontario Ministry of Health and Long-Term Care. Gabor Fichtinger was funded as a Cancer Ontario Research Chair. Tamas Ungi was supported as a Queen's University—Ontario Ministry of Research and Innovation Postdoctoral Fellow. Laura Bartha was supported by the Human Mobility CREATE program of The Natural Sciences and Engineering Research Council of Canada.

**Conflict of interest** None.

## References

- Gill S, Abolmaesumi P, Fichtinger G, Boisvert J, Pichora D, Borshneck D et al (2010) Biomechanically constrained groupwise ultrasound to CT registration of the lumbar spine. *Med Image Anal* 16(3):662–674
- Gjerald SU, Brekken R, Hergum T, D'hooge J (2012) Real-time ultrasound simulation using the GPU. *IEEE Trans Ultrason Ferroelectr Freq Control* 59(5):885–892
- Goksel O, Salcudean SE (2009) B-Mode ultrasound image simulation in deformable 3-D medium. *IEEE Trans Med Imaging* 28(11):1657–1669
- Hedrick WR, Hykes DL, Starchman DE (2005) *Ultrasound physics and instrumentation*. Mosby, St. Louis
- Hostettler A, Forest C, Forgione A, Soler L, Marescaux J (2005) Real-time ultrasonography simulator based on 3D CT-scan images. *Stud Health Technol Inform* 111:191–193
- Jensen JA (1996) FIELD: a program for simulating ultrasound systems, pp 351–353
- Jensen J (2004) Simulation of advanced ultrasound systems using Field II, vol 1. pp 636–639
- Karamalis A (2009) GPU ultrasound simulation and volume reconstruction. Master's thesis, Technischen Universität München



9. Karamalis A, Wein W, Navab N (2010) Fast ultrasound image simulation using the Westervelt equation. In: Medical image computing and computer-assisted intervention-MICCAI 2010. Springer, Berlin, pp 243–250
10. Kutter O, Shams R, Navab N (2009) Visualization and GPU-accelerated simulation of medical ultrasound from CT images. *Comput Methods Programs Biomed* 94(3):250–266
11. Lasso A, Heffter T, Pinter C, Ungi T, Fichtinger G (2012) Implementation of the PLUS open-source toolkit for translational research of ultrasound-guided intervention systems. In: Medical image computing and computer-assisted intervention (MICCAI) workshop—Systems and architectures for computer assisted interventions, Nice, France, The MIDAS Journal, pp 1–12, 10/2012
12. Lasso A, Heffter T, Pinter C, Rankin A, Ungi T, Fichtinger G (in review) PLUS: open-source toolkit for ultrasound-guided intervention systems. *IEEE Trans Med Imaging*
13. Magee D, Zhu Y, Ratnalingam R, Gardner P, Kessel D (2007) An augmented reality simulator for ultrasound guided needle placement training. *Med Biol Eng Comput* 45(10):957–967
14. Moulton E, Ungi T, Welch M, Lu J, McGraw RC, Fichtinger G (2013) Ultrasound-guided facet joint injection training using Perk Tutor. *Int J Comput Assist Radiol Surg*. doi:10.1007/s11548-012-0811-5
15. Pieper S, Lorensen W, Schroeder W, Kikinis R (2006, 04) The NA-MIC Kit: ITK, VTK, pipelines, grids and 3D Slicer as an open platform for the medical image computing community, pp 698–701
16. Schroeder W, Martin K (2004, 01) The visualization Toolkit
17. Skehan D (2011) Virtual training system for diagnostic ultrasound. Master's thesis, Worcester Polytechnic Institute, The United States of America
18. Tokuda J, Fischer G, Papademetris X, Yaniv Z, Ibanez L, Cheng P et al (2009) OpenIGTLink: an open network protocol for image-guided therapy environment. *Int J Med Robotics Comput Assist Surg* 5(4):423–434
19. Ungi T, Sargent D, Moulton E, Lasso A, Pinter C, McGraw R et al (2012) Perk Tutor: an open-source training platform for ultrasound-guided needle insertions. *IEEE Trans Biomed Eng* 59(12):3475–3481
20. Vidal FP, John NW, Healey AE, Gould DA (2008) Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics. *Comput Animat Virtual Worlds* 19(2):111–127
21. Wein W, Khamene A, Clevert D-A, Kutter O, Navab N (2007) Simulation and fully automatic multimodal registration of medical ultrasound. In: Medical image computing and computer-assisted intervention—MICCAI 2007. Springer, Berlin, pp 136–143
22. Zhu M, Salcudean S (2010) Real time ultrasound needle image simulation using multi-dimensional interpolation. In: Jiang, T, Navab, N, Pluim J, Viergever, M (eds) Medical image computing and computer-assisted intervention “ MICCAI 2010”, vol 6362. Springer, Berlin, pp 429–436