

# Real-Time Synthesis of Image Slices in Deformed Tissue from Nominal Volume Images

Orcun Goksel and Septimiu E. Salcudean

Department of Electrical and Computer Engineering  
University of British Columbia, Vancouver, Canada  
{orcung,tims}@ece.ubc.ca

**Abstract.** This paper presents a fast image synthesis procedure for elastic volumes under deformation. Given the node displacements of a mesh and the 3D image voxel data of an undeformed volume, the method maps the image plane pixels to be synthesized from the deformed configuration back to the nominal pre-deformed configuration, where the pixel intensities are obtained easily through interpolation in the regular-grid structure of the voxel volume. For smooth interpolation, this mapping requires the identification of the mesh element enclosing each image pixel. To accelerate this *point location* procedure, a fast method of marking the image pixels is employed by finding the intersection of the mesh and the image, and marking this intersection on the image pixels using *Bresenham's line drawing algorithm*. A deformable tissue phantom was constructed, it was modeled using the finite element method, and its 3D ultrasound volume was acquired in its undeformed state. Actual B-mode images of the phantom under deformation by the ultrasound probe were then compared with the corresponding synthesized images simulated for the same deformations. Results show that realistic images can be synthesized in real-time using the proposed technique.

## 1 Introduction

Medical simulators involving real-time imaging modalities, such as ultrasound, necessitate rapid and realistic image rendering of deformed tissue in response to probe or tool manipulation by a trainee. For real-time performance, tissue deformation, typically modeled by the finite element method (FEM), must be computed on a mesh having much coarser elements than the typical resolution of medical imaging modalities. Exploiting this fact, our work addresses the computational problems related to rapidly slicing a deformed 3D mesh. The need for real-time image synthesis in ultrasound simulation makes it our primary target application, although the techniques presented here are also applicable to other medical imaging modalities that need to slice deformed meshes.

Image generation inside deformed tissue has some concepts in common with the *elastic image registration* [1,2] and the *ultrasound volume reconstruction* [3,4,5] literature, where reconstruction refers to generating 3D voxel volume data from individual scans. However, compared to these fields that generally process

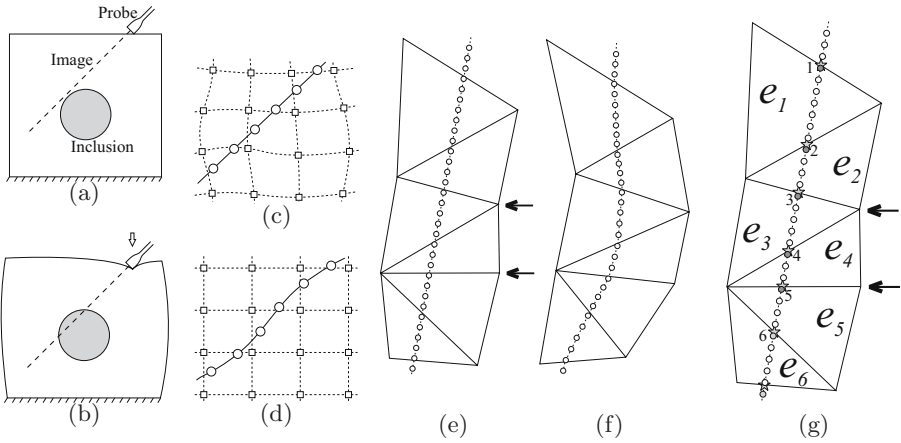
images offline, real-time medical image generation presents additional computational challenges, which have been studied for ultrasound B-mode image simulation in the literature. There exist two approaches for ultrasound image synthesis, *the generative approach* and *the interpolative approach*. The former models the ultrasonic wave propagation by using accurate representations of the probe, the tissue scatterers, and the wave interaction [6]. This complex and time-consuming approach is not suitable for real-time applications. The latter approach slices images from a pre-existing voxel volume data. UltraSim [7] and several others [8,9] follow this latter approach. However, these ultrasound simulators do not allow for tissue deformation.

In many medical procedures such as prostate brachytherapy, brain surgery, or breast biopsy, significant deformation is caused by medical tools or by the ultrasound probe. In certain applications, such as diagnosis of deep-vein thrombosis (DVT), deformation observed in ultrasound images during deliberate probe indentation contains essential diagnosis information. Fast synthesis of ultrasound images in soft tissues under deformation will facilitate the development of training simulators. With this goal, a DVT diagnosis simulator was proposed in [10,11]. It simulates the probe pressure by first slicing an image from the 3D ultrasound data set and then applying a 2D in-plane elastic deformation using quadtree-splines to this image. This 2D deformation applied is pre-computed offline by registering segmented anatomy from undeformed to deformed 3D models. However, an in-plane image deformation approach is not capable of simulating situations such as in Fig. 1(a-b), where anatomical structures enter or leave the imaging plane due to tissue deformation. In the example depicted, an inclusion which is nominally not in the ultrasound imaging plane later appears in it as a result of the depression by the probe. Real-time ultrasound image slicing using physically-valid 3D deformation models has not been addressed in the literature. Our work is motivated by this need for real-time realistic B-mode ultrasound simulation in medical training environments involving tissue deformation.

The paper is organized as follows. First, the overall interpolation approach of mapping image pixels back to the undeformed volume is introduced. Then, our acceleration technique for this approach is presented and it is applied to real-time ultrasound synthesis of a gelatin phantom while modeling deformation caused by the ultrasound probe, similarly to [12]. In this paper, deformed images of the phantom are acquired and compared with the simulated ones quantitatively. The change of the algorithm processing time with image size is also studied. Complementary detail on the specific numerical techniques presented below can be found in [12].

## 2 Methods

The techniques described below assume that a reconstructed voxel volume data of the region of interest is available *a priori*. Consider the deformed tissue configuration in Fig.1(c). To generate an image given the *reconstructed* voxel data, the intensity values (*gray-values*) at the planar locations shown need to be



**Fig. 1.** A sample case where in-plane image deformation cannot capture the outcome of volume deformation from (a) nominal to (b) deformed configurations; Voxel data and image plane in (c) deformed and (d) undeformed states (the circles are the image pixels and the squares are the reconstructed data voxels); (e-f) mesh representation of an undeformation (reconstructed voxels are not depicted in here); and (g) flagging of pixels in a deformed 2D mesh

interpolated. Note that interpolating in the deformed data in the configuration given in Fig. 1(c) has two problems. First, the data is no longer in a regular-grid structure, requiring computationally-expensive scattered-data interpolation techniques. Second, although only a small subset of the nominal voxels are needed in the interpolation, it is time consuming to find which voxels are required. One can instead apply the inverse of the (deformation) displacement on image pixels in order to transform them back (*undeform*) into the regular-grid structure of the voxel volume as in Fig. 1(d). This latter approach is utilized in this paper.

Tissue tessellations for deformation models are generally much coarser than the imaging resolution due to computational constraints. Consider the situation in Fig. 1(e) where an image plane slices an object deformed under boundary constraints shown with the arrows as simulated by linear FEM using the given mesh. Mapping the mesh back to its nominal configuration (*undeforming*) as shown in Fig. 1(f) brings the image pixels to locations at which the given regular-grid data can be interpolated easily. Consequently, the simulation scheme for each pixel position  $P$  can be summarized as (i) finding its enclosing element  $e_i$ , (ii) computing the *undeforming* transformation  $T$  using  $e_i$ 's shape functions, and (iii) interpolating the transformed position  $TP$ . Identifying the element enclosing a pixel in (i) is the bottleneck of this approach, since (ii) and (iii) are fast constant time  $\mathcal{O}(1)$  operations. Note that this pixel-element mapping depends on both mesh deformation and the image position/orientation; therefore, it cannot be computed offline. This problem, called *point location* in computational geometry, has been extensively studied resulting in common techniques such as *slab decomposition* and *trapezoidal maps*. Nonetheless, execution

time of any such method will not allow for real-time processing of conventional image resolutions. Indeed, the point location routine in QuickHull algorithm [13] locates the pixels of a single image presented in Section 3 in over 30 s. Instead of running point location individually for each pixel, in this paper, we exploit the fact that all the pixels lie on a planar image having finer resolution than the tissue tessellation.

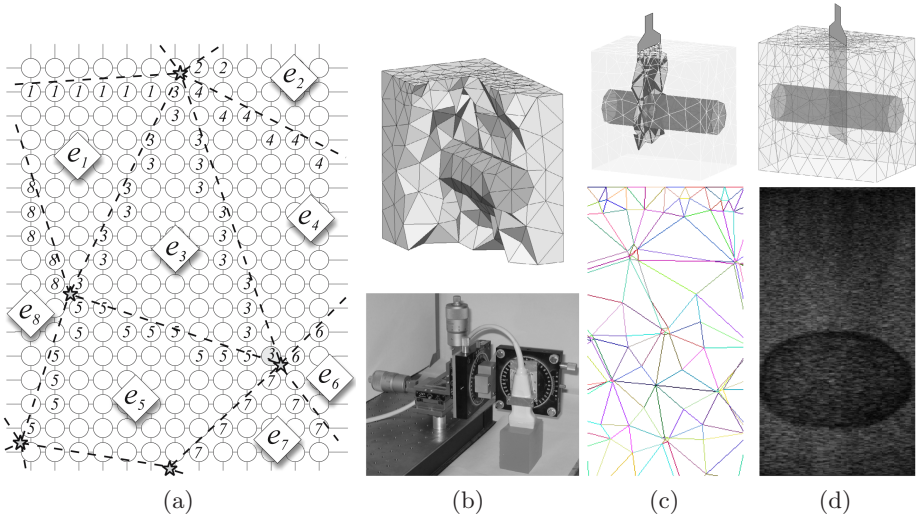
Note that when pixels in an image are traversed in one direction, the elements that they belong to only alternate at the element edges. This fact can be utilized to rapidly identify the enclosing elements of pixels. For example, in a 2D mesh as in Fig. 1(g), if the pixels just below the upper-edge of each element are flagged by that element number, then a downwards traversal will reveal the elements enclosing the rest of the pixels in constant time. Similarly, in 3D this corresponds to flagging all the pixels along the entire surface cross-section of *sliced elements* (any element that is intersected by the image being generated), since these cross-sections are the borders where the assigned elements for consecutive pixels change. Accordingly, for each frame, a pre-computation of pixel flagging is introduced in order to reduce the step (i) above to negligible time.

Figure 2(a) demonstrates a sample flagging array slicing a geometrically-linear tetrahedral mesh. A plane can slice either 3 or 4 edges of a tetrahedron. Simple geometrical computations to find these intersections shown with stars require negligible  $\mathcal{O}(1)$  time per sliced element. Then, a cross-section can be flagged by discretizing the lines connecting these intersections onto the flagging array. For this we use an optimized version of *Bresenham's line-drawing algorithm*. This only requires the assumption that the image pixels are equally-spaced in each individual axis. Assuming a top-down array traversal, only the top halves of triangles/quadrilaterals are flagged.

Let  $s$  be the number of sliced elements, then an average of  $\sqrt{s}$  elements span one axis of the image, each causing  $\sqrt{n}$  pixels to be flagged on average. This gives the cost  $\mathcal{O}(\sqrt{ns})$  for preparing this flagging array prior to the generation of each frame. Since the frame interpolation still requires the traversal of  $\mathcal{O}(n)$  pixels, the overall computational complexity is not reduced technically. Nonetheless, removing the significant hidden cost of point location from the  $\mathcal{O}(n)$  loop practically achieves the anticipated speed gain.

### 3 Results

To obtain 3D ultrasound voxel data, a  $60 \times 90 \times 90$  mm gelatin phantom was prepared. For contrast in ultrasound, a softer cylindrical inclusion of  $\phi 25$  mm was made by varying the cellulose scatterer content. The phantom mesh, obtained using off-the-shelf meshing software, has 493 nodes and 1921 tetrahedra. For simulating deformation, considering the approximate values known for the materials, Young's modulus of the inclusion is set twice the substrate and a Poisson's ratio of 0.49 is used for both materials. Parallel slices of this volume were collected at 1 mm increments, paying attention to minimally deforming the phantom, using the Ultrasonix Sonix RP machine with a linear probe mounted on a precision

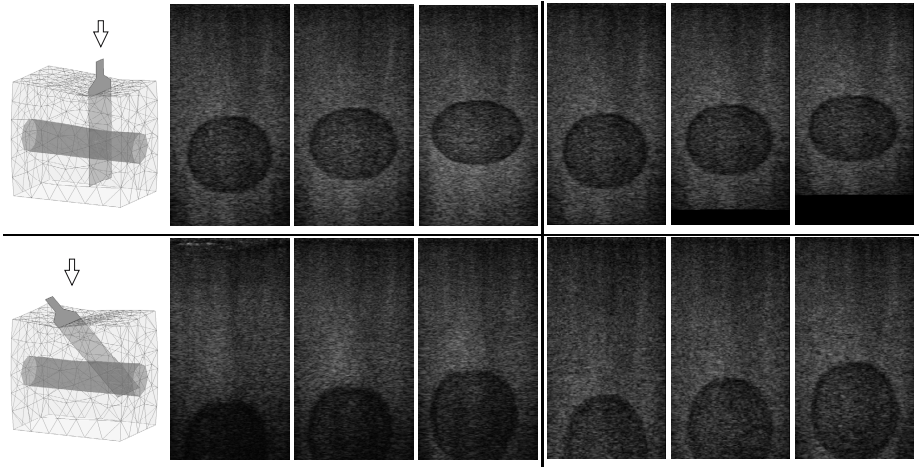


**Fig. 2.** (a) Flagging a sample 3D mesh projection on an array; (b) our phantom mesh and the image acquisition setup; (c) the cross-sections of the elements traversed and their boundaries; and (d) a synthesized image in tissue without any deformation

motion stage. The geometry and mesh of the phantom, and the setup for image acquisition can be seen in Fig. 2(b). Each image spans  $37.5 \times 70$  mm and has a resolution of  $220 \times 410$ . This is the typical B-mode resolution on this ultrasound machine. This dense data is then used as the reconstructed voxel volume without any further interpolation. In this configuration, there is an average of 8750 voxels per element.

Figure 2(c) shows the cross-sections of traversed elements and the corresponding flagging array. A sample image without any phantom deformation (probe pressure) is shown in Fig. 2(d). To simulate deformation, the bottom layer of the phantom is fixed and the mesh nodes touching the probe on the top surface were displaced to conform to the probe. Two predetermined regions on the phantom had already been meshed with finer elements having nodes lying on the anticipated interfaces. In Fig. 3 the images were simulated as the probe indents 0, 5, and 10 mm into the phantom at two given positions/orientations. To measure the effect of  $n$  and  $s$ , the slice generation time of our implementation on a Pentium4 2.4 GHz computer was compared for various image *resolutions* and *widths* in Fig. 4(a). Note that the former only changes  $n$ , whereas the latter also affects  $s$ . The linearity and similarity of the lines show that generating flagging array takes negligible time and  $\mathcal{O}(n)$  is the dominant factor. Thus, our approach of optimizing its hidden cost is an effective strategy for accelerating this scheme.

Simulated and acquired images at 1 mm indentation increments during the deformation in Fig. 3(top) were compared using their *mutual information* (MI) in Fig. 4(b). Plotted MI data were normalized with the average MI between each consecutive slice acquired for reconstruction throughout the volume. As



**Fig. 3.** Acquired (center) and simulated (right) images at 0, 5, and 10 mm indentation depths with vertical (top) and 45°-tilted (bottom) probe orientations (meshes (left) are depicted for 10 mm only)

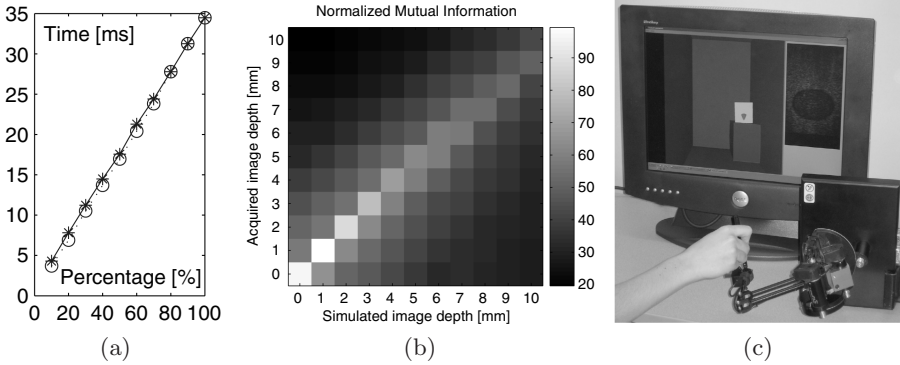
expected, MI is the highest on the diagonal (i.e., each simulated image is most alike the acquired one at a similar indentation). A preliminary system for interactive ultrasound visualization with deformation has been developed using a SensAble Phantom interface as seen in Fig. 4(c). A simplified contact handling method of applying the deformation only to the closest node was implemented. A depth-dependent reaction force simulated with a spring from the nominal surface is applied on user's hand.

## 4 Discussion

In the literature, there have been image slicing implementations without taking deformation into account [5,7,8,9]. Also, there have been *in-plane image deformation* strategies for image registration, deformation correction for volume reconstruction [3,4], and a training simulation for DVT [10,11]. However, to the extent of our knowledge, our implementation is the first real-time image slicer inside 3D deformation models. Although only the probe indentation was presented here, the source of deformation can be also external such as needles or other medical instruments. In some procedures, e.g. prostate brachytherapy, the major tissue deformation is orthogonal to the imaging plane. In such cases, 2D in-plane deformation simply cannot be used and our method becomes essential.

Our acceleration techniques for this image synthesis method do not introduce any additional errors beyond the FEM simulation errors, i.e., we do not trade off accuracy with speed. As one will expect, for significantly large deformations, the acquired and the simulated images may not match exactly due to errors in the deformation model and differences between the estimated and the actual elasticity parameters.





**Fig. 4.** (a) Slice generation time for various image resolutions (stars) and widths (circles) normalized to the full-scale size of the images presented above; (b) Mutual information in-between simulated and acquired images for vertical indentation at 1 mm increments; and (c) our real-time ultrasound examination simulator

We employ the linear FEM with tetrahedra for deformation due to its *shape functions* being simple, continuous, and easily invertible. Nonetheless, all presented methods also apply to the FEM with quadrilateral or other-geometry elements. Note that in flagging arrays, some pixels close to element corners/edges may be assigned to more than one element due to discretization. In order to successfully flag them, sliced polygons are first topologically sorted in linear time using their top-down partial ordering.

Due to speckle and other directional imaging artifacts, a deformed tissue does not necessarily generate the exact same nominal gray-values at its displaced position. Nevertheless, this assumption has been employed in many applications in the literature [3,4,5,7,8,9,10,11], with success. The speckle pattern (mainly its visual continuity) is also a major criterion in evaluating the realism of synthesized images. In this respect, the generated images were qualitatively found to be adequate for a training simulator.

## 5 Conclusions

We have presented a technique to synthesize planar images from the deformed mesh of a tissue model and the 3D image voxel data of the undeformed tissue. The deformation is assumed to be given by the node displacements of a mesh. A fast processing step was developed to identify the mesh element to which each pixel belongs by employing a plane enumeration technique. This allows the generation of image planes of considerable size at frame rates that are suitable for real-time applications. A phantom was constructed and its 3D B-mode ultrasound image was collected. The phantom was meshed and its deformation due to probe indentation was simulated using the FEM. Planar ultrasound images of the deformed phantom were synthesized and compared to the corresponding images acquired by deforming the phantom. The results show that the image

synthesis method we developed produces realistic-looking images in real-time. In future work, we will target specific clinical applications in which we will carry out more extensive evaluation.

## References

1. Krücker, J.F., LeCarpentier, G.L., Fowlkes, J.B., Carson, P.L.: Rapid elastic image registration for 3-D ultrasound. *IEEE Trans. Med. Imag.* 21(11), 1384–1394 (2002)
2. Zikic, D., Wein, W., Khamene, A., Clevert, D.A., Navab, N.: Fast deformable registration of 3D-ultrasound data using a variational approach. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) *MICCAI 2006*. LNCS, vol. 4190, pp. 915–923. Springer, Heidelberg (2006)
3. Burcher, M.R., Han, L., Noble, J.A.: Deformation correction in ultrasound images using contact force measurements. In: *IEEE Workshop Math Methods Bio Img Anal*, Kauai, HI, USA, pp. 63–70 (2001)
4. Rohling, R.N., Gee, A.H., Berman, L.: A comparison of freehand three-dimensional ultrasound reconstruction techniques. *Med. Image. Anal.* 3(4), 339–359 (1999)
5. Gee, A., Prager, R., Treece, G., Cash, C., Berman, L.: Processing and visualizing three-dimensional ultrasound data. *British J. Radiology* 77, S186–S193 (2004)
6. Jensen, J.A., Nikolov, I.: Fast simulation of ultrasound images. In: *IEEE Ultrasonics Symposium* pp. 1721–1724 (2000)
7. Aiger, D., Cohen-Or, D.: Real-time ultrasound imaging simulation. *Real-Time Imaging* 4(4), 263–274 (1998)
8. Maul, H., Scharf, A., Baier, P., Wüstemann, M., Günter, H.H., Gebauer, G., Sohn, C.: Ultrasound simulators: Experience with sonotrainer and comparative review of other training systems. *Ultrasound Obstet Gynecol* 24, 581–585 (2004)
9. Tahmasebi, A.M., Abolmaesumi, P., Hashtrudi-Zaad, K.: A haptic-based ultrasound training/examination system (HUTES). In: *IEEE ICRA*, Roma, Italy, pp. 3130–3131 (2007)
10. Henry, D., Troccaz, J., Bosson, J.L., Pichot, O.: Ultrasound imaging simulation: Application to the diagnosis of deep venous thromboses of lower limbs. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) *MICCAI 1998*. LNCS, vol. 1496, pp. 1032–1040. Springer, Heidelberg (1998)
11. d’Aulignac, D., Laugier, C., Troccaz, J., Vieira, S.: Towards a realistic echographic simulator. *Med. Image. Anal.* 10, 71–81 (2005)
12. Goksel, O., Salcudean, S.E.: Fast B-mode ultrasound image simulation of deformed tissue. In: *IEEE EMBC*, Lyon, France (2007)
13. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.T.: The Quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22(4), 469–483 (1996)