A Framework for GPU Accelerated Needle Insertion Simulation using Meshfree Methods

Aria Shahingohar*
University of Western Ontario

Roy Eagleson†
University of Western Ontario

Abstract

The simulation of needle insertion is an important research area that has many applications in robotic and image guided brachytherapy cancer treatment, biopsies, and neurosurgery. Modeling of soft tissue plays an important role in the needle insertion simulation, but the use of Finite Element Method is complicated due to the need for remeshing in the neighbourhood of the needle tip. We are proposing to use a meshfree method for the tissue deformation modeling, in which new tissue nodes are added on the needle shaft as the needle is inserted into the tissue. In addition, we have utilized Nvidia's CUDA technology to accelerate the methods used in our framework.

Keywords: Deformable Object, Needle Insertion Simulation, Meshfree Methods

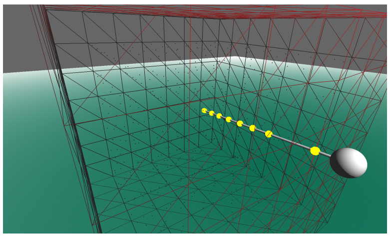

Figure 1: *A screenshot of needle insertion simulation in our framework.*

Our Framework

Zhu et. al.[Zhu et al. 2007] uses a localized mass-spring model to simulate needle tissue interaction. They assume that the needle only causes deformation in the region local to the insertion path. Therefore, once the needle is inside, a cylindrical deforming field centered at the predicted insertion path (i.e., the current direction of the needle) is created in the form of a tetrahedral mesh.

In [Chentanez et al. 2009] the Finite Element Method is used to model tissue deformations during needle insertion and the *condensation* method is used to achieve higher frame rates. However, there is a fundamental challenge when simulating needle insertion with FEM and that is the fact that boundary conditions can only be applied on the nodes. Since the needle tip can pass through elements, the tissue should be constantly remeshed around the tip to ensure it is always aligned with a node. Remeshing is complicated and costly, especially if the quality of the new mesh should be preserved.[Chentanez et al. 2009]

In our proposed framework, meshfree methods are used for the simulation. Our main contribution is that instead of remeshing or snapping the nodes, we add nodes to the system. Adding a node is done in such a way that it does not impose any artificial strain in the tissue.

In our framework, for each node a set of arrays are generated to store several variables associated with the simulation. These variables are stored for all nodes and include the position, velocity, elastic force, list of neighbours and the weights of each neighbour. The original positions of the nodes are also kept since it is required in most methods. If we decide to add a node to our model (for example if the needle is penetrated more than a threshold) we have to make a list of neighbours for it and calculate the weight for each of its neighbour nodes. Moreover, the new node should be added to the list of neighbours of all nodes in its vicinity and the corresponding weight should also be updated. On the other hand, in each meshfree method some values should be calculated on the original undeformed shape. For example in [Müller et al. 2004] $\mathbf{A}^{-1} = (\sum_j \mathbf{x}_{ij}\mathbf{x}_{ij}^T w_{ij})^{-1}$ is pre-computed for all the nodes at the beginning of the simulation. This value should be calculated for the newly added node and it should be updated for all other affected nodes (neighbours of the new node). In order to recalculate these values, the position of the newly added node in the reference mesh is needed. The position of the newly added node in the reference mesh(or the material space) is obtained by an approximation based on the position of the new node in the deformed state.

In order to be able to remove the added nodes due to pulling the needle out, two stacks are defined to hold the node neighbours and node weights arrays. Every time a node is added, current arrays are pushed into the stack before updating, and the stack is popped if the last added node is deleted.

We have successfully modeled a non-deflecting needle with this technique. We use a stick slip method for the friction between needle shaft and nodes. The newly added nodes are constrained to move with the needle in the stick state. A node goes to the slip mode if the elastic force on it exceeds a threshold. In the slip mode boundary condition is enforced by projecting each node to the needle shaft trajectory.

Since most meshfree methods can be performed faster if they are executed in parallel, our framework utilizes Nvidia's CUDA technology to accelerate the calculations by transferring their core computations to the GPU. Our results show that frame rates can be improved to more than 20 times using GPU.

References

CHENTANEZ, N., ALTEROVITZ, R., RITCHIE, D., CHO, L., HAUSER, K. K., GOLDBERG, K., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2009. Interactive simulation of surgical needle insertion and steering. In *Proceedings of ACM SIGGRAPH 2009*.

MÜLLER, M., KEISER, R., NEALEN, A., PAULY, M., GROSS, M., AND ALEXA, M. 2004. Point based animation of elastic, plastic and melting objects. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 141–151.

ZHU, Y., MAGEE, D. R., RATNALINGAM, R., AND KESSEL, D. 2007. A training system for ultrasound-guided needle insertion procedures. In *MICCAI (1)*, 566–574.

*e-mail: ashahing@uwo.ca
†e-mail:eagleson@uwo.ca

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007