Graphics Processing Units for the Real-time Linear Elastostatic Simulation of Liver

Kirana Kumara P, Ashitava Ghosal
Centre for Product Design and Manufacturing
Indian Institute of Science
Bangalore, India
kiran_kp1@rediffmail.com

Abstract—**Biomedical engineering solutions like surgical simulators need High Performance Computing (HPC) to achieve real-time performance. Graphics Processing Units (GPUs) offer HPC capabilities at low cost and low power consumption. In this work, it is demonstrated that a liver which is discretized by about 2500 finite element nodes, can be graphically simulated in real-time, by making use of a GPU. Present work takes into consideration the time needed for the data transfer from CPU to GPU and back from GPU to CPU. Although behaviour of liver is very complicated, present computer simulation assumes linear elastostatics. One needs to use the commercial software ANSYS to obtain the global stiffness matrix of the liver. Results show that GPUs are useful for the real-time graphical simulation of liver, which in turn is needed in simulators that are used for training surgeons in laparoscopic surgery. Although the computer simulation should involve rendering also, neither rendering, nor the time needed for rendering and displaying the liver on a screen, is considered in the present work. The present work is just a demonstration of a concept; the concept is not really implemented and validated. Future work is to develop software which can accomplish real-time and very realistic graphical simulation of liver, with rendered image of liver on the screen changing in real-time according to the position of the surgical tool tip approximated as the mouse cursor in 3D.**

Keywords-GPU; simulation; liver; real-time; FEM

I. INTRODUCTION

Laparoscopic surgery is a minimally invasive surgery and needs highly skilled surgeons. It is risky for the surgeons to acquire these skills by performing surgeries on real patients. Hence a surgical simulator is needed to train surgeons in this kind of surgeries.

Surgical simulators need virtual biological organs. A surgical simulator that can simulate laparoscopic surgery needs a virtual organ such as a liver. The organ, a liver for the present work, has to be simulated in real-time.

In a real laparoscopic surgery, surgeons do not receive much force feedback; they rely on the graphical display of the organ that is displayed on a monitor. Hence, in a laparoscopic surgery simulator, it is not essential to simulate forces; but one has to graphically simulate the liver and these simulations have to be carried out at real-time. A real-time graphical simulation needs to update graphics at about thirty times a second [1], and hence the whole of computations that are needed to obtain real-time graphics can take about one by thirtieth of a second only.

The necessity of completing the complicated computations within the allowed time forces one to take either of the following two approaches. The first approach achieves the speed by compromising on the realism altogether, by adopting models that are not physically based. The second approach compromises on the speed (which effectively compromises on the realism also) but adopts complicated models that are physically based. Present work takes somewhat an intermediate path and assumes the liver to be linear elastic. Although the linear elastostatic model cannot accurately describe liver, the model is based on continuum mechanics, and for better realism, models based on continuum mechanics are preferred [2]. Also, adopting linear elastostatics for the present work has enabled the computations to be carried out at real-time.

Present work uses a GPU to accelerate the computations. When compared to CPUs, GPUs offer the same parallel computing performance for lesser cost and lower power consumption. But, presently, not all kinds of problems can benefit from the parallel computing capabilities offered by GPUs. However, GPUs have already established themselves in some areas like matrix multiplication. In fact, present work uses a GPU to accelerate only the matrix multiplication part of the computations.

II. NUMERICAL MODELING

The liver geometry is extracted from the CT-scan data from [3]. Data was downloaded some time back when it was available in the website, but now, the website does not contain the scan data. The procedure followed to get the three-dimensional geometry of the porcine liver is explained in [4]. The liver is shown in Fig. 1 ("Figure 8. The Liver Displayed in MeshLab (102 Vertices or 200 Faces; Well Shaped Triangles)," Reproduced from [4]).

The 3D surface model (shown in Fig. 1) may be converted to a 3D solid model using a software such as Rhinoceros, as has been indicated in [4]. Reference [4] also demonstrates that the models obtained by following the procedure given in [4], do not pose difficulty during meshing. In fact, [4] also demonstrates that the models produced by following the procedure given in [4] may be readily used in an analysis.

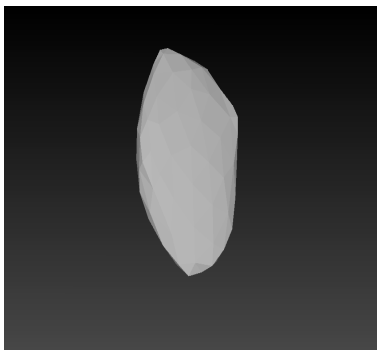

Figure 1. 3D model of liver

Now, linear elastostatics is described by the following governing equations [5]:

$$(\lambda+\mu)\left\{\frac{\partial^2 u_x}{\partial x^2}+\frac{\partial^2 u_y}{\partial x\partial y}+\frac{\partial^2 u_z}{\partial x\partial z}\right\}+\mu\left\{\frac{\partial^2 u_x}{\partial x^2}+\frac{\partial^2 u_x}{\partial y^2}+\frac{\partial^2 u_x}{\partial z^2}\right\}=0 \quad (1)$$

$$(\lambda+\mu)\left\{\frac{\partial^2 u_y}{\partial y^2}+\frac{\partial^2 u_x}{\partial x\partial y}+\frac{\partial^2 u_z}{\partial y\partial z}\right\}+\mu\left\{\frac{\partial^2 u_y}{\partial x^2}+\frac{\partial^2 u_y}{\partial y^2}+\frac{\partial^2 u_y}{\partial z^2}\right\}=0 \quad (2)$$

$$(\lambda+\mu)\left\{\frac{\partial^2 u_z}{\partial z^2}+\frac{\partial^2 u_x}{\partial x\partial z}+\frac{\partial^2 u_y}{\partial y\partial z}\right\}+\mu\left\{\frac{\partial^2 u_z}{\partial x^2}+\frac{\partial^2 u_z}{\partial y^2}+\frac{\partial^2 u_z}{\partial z^2}\right\}=0 \quad (3)$$

where $\lambda=\frac{\nu E}{(1+\nu)(1-2\nu)}$ $\mu = \frac{E}{2(1+\nu)}$

Here, u_x, u_y and u_z are the displacements along x, y and z directions; E is the Young's modulus and ν is the Poisson's ratio.

The Finite Element Method (FEM) [6], a well known numerical technique, may be used to numerically model the liver. The commercial software ANSYS [7] may be used for this purpose. Role of ANSYS here is to generate the characteristic matrix (or the global stiffness matrix) for the liver. One can get all components of the global stiffness matrix of the model of the liver by converting the model to an ANSYS "Superelement". All degrees of freedom should be considered "Master Degrees of Freedom". The process does not require any boundary conditions, and the "Analysis Type" should be chosen as "Substructuring".

One can satisfactorily describe a liver by about 2500 nodes, e.g., in [8], liver has been described by even lesser points. Present work assumes that the discretization is carried out such that the total number of nodes equals 2500. There are three degrees of freedom (u_x, u_y and u_z) at each of the nodes. Hence the size of the global stiffness matrix is (7500 by 7500).

III. Solution Strategy

The problem is to find the solution [X] for the matrix equation:

$$[K]\{X\} = \{F\} \quad (4)$$

where [K] is the global stiffness matrix,
{X} is the solution that represents displacement components (u_x, u_y and u_z) at each of the nodes in the finite element discretization
{F} is the global force vector

Equation (4) may be written as:

$$\{X\}=[K]^{-1}\{F\} \quad (5)$$

Equation (5), together with boundary conditions, may be solved using MATLAB [9]. A Graphics Processing Unit (GPU) may be used to accelerate the computations. The GPU used here is NVIDIA GeForce GTX 460. GPU is accessed from within MATLAB using GPUmat [10], a GPU toolbox for MATLAB. Fig. 2 illustrates the solution strategy.

During each iteration, one needs to transfer the global force vector {F} from CPU to GPU, compute {X} from (5) in GPU, and transfer the calculated {X} from GPU to CPU. Real-time graphical simulation requires all of these to be completed within 1/30th of a second (0.033 second). For linear elastostatic problems, the global stiffness matrix [K] does not vary over time and hence the inverse of [K] can be precomputed and transferred to GPU before any iteration begins. This is to minimize the data transfer between CPU and GPU since it is a time consuming process.

IV. Results

One iteration with only CPU takes 0.066 seconds to complete. One iteration with GPU, including transferring {F} from CPU to GPU and transferring {X} from GPU to CPU, takes 0.030 seconds to complete. Although, during each iteration, one should consider the time for transferring {F} from CPU to GPU and transferring {X} from GPU to CPU, if

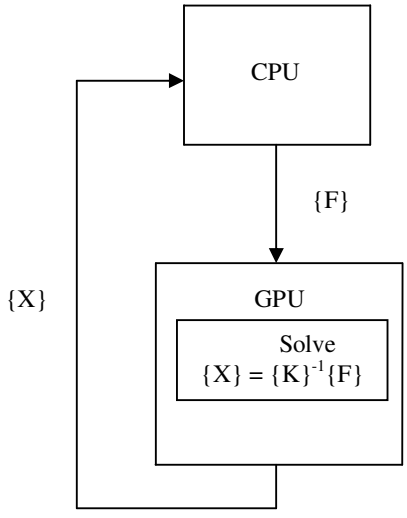

Figure 2. Solution strategy

one does not take into consideration these data transfer times, only the time for solving (5) on the GPU is 0.020 seconds.

For models other than linear elastostatics, [K] can vary over time and it may also be necessary to transfer [K] during each iteration. Although this is not the case with the present work, one can see that the transfer of [K] alone from CPU to GPU takes 0.184 seconds. This clearly shows that the data transfer between CPU and GPU is a very time consuming task and while implementing nonlinear models (not while implementing linear elastostatics), to take advantage of GPU, one should be very careful to design and implement customized and efficient GPU algorithms that minimize the data transfer between CPU and GPU.

The above results are based on the time required for multiplying a 7500 by 7500 matrix with a 7500 by 1 vector. It is true that after incorporation of boundary conditions, the dimensions of matrices reduce a bit. But, it can be seen that this does not noticeably change solution times.

V. CONCLUDING REMARKS

From the results, one can see that even with the linear elastostatic idealization of the liver, one cannot complete the computations within the allowed time of one by thirtieth of a second if one uses a CPU alone. But one can see that, with the GPU acceleration, one can graphically simulate the liver at real-time. GPU being a low cost High Performance Computing (HPC) solution, and since it requires lesser power keeping in mind its computing capabilities, GPUs are ideal for use in applications such as laparoscopic surgical simulators where enormous amount of computations have to be performed within a very short period of time.

In the present work, 'CPU' refers to Central Processing Unit minus GPU. Although GPUs are usually fitted within Central Processing Units, 'CPU' in the present work does not include any GPU. A single Central Processing Unit (not a cluster) but with six processors inside, is the 'CPU' used for the present work. 'GPU' refers to the single NVIDIA GPU.

Using languages such as C and GLSL instead of MATLAB and GPUmat for the present work might have reduced solution times but this has not been tried out in the present work. Also, in this work, rendering and graphically displaying the liver on a screen during the real-time simulation, is not considered; not even the time required for this rendering and displaying is taken into account. Further, whole of the present work is just a demonstration of only a concept; the concept has not been implemented and validated.

Future work is to incorporate more realistic models instead of the linear elastostatics. Aim is to obtain real-time performance with these nonlinear models by making use of a heterogeneous system consisting of multiple CPUs/clusters and GPUs. One has to inevitably consider rendering also. Also, the concept which makes up the present work has to be really implemented and validated. Ultimate goal is to develop software which can accomplish real-time and very realistic graphical simulation of liver, with rendered image of the liver on the screen changing in real-time according to the position of the surgical tool tip approximated as the mouse cursor in 3D.

REFERENCES

[1] OPENHAPTICS™ TOOLKIT version 2.0, PROGRAMMER'S GUIDE, SensAble technologies, 2005

[2] U. Meier, O. Lopez, C. Monserrat, M. C. Juan, M. Alcaniz, "Real-time deformable models for surgery simulation: a survey," Computer Methods and Programs in Biomedicine (2005) 77, 183-97

[3] (2010) [Online]. Available: http://biorobotics.harvard.edu

[4] Kirana Kumara P, Ashitava Ghosal, "A Procedure for the 3D Reconstruction of Biological Organs from 2D Image Sequences," Proceedings of BEATS 2010, 2010, International Conference on Biomedical Engineering and Assistive Technologies (BEATS 2010), Dr B R Ambedkar National Institute of Technology, Jalandhar

[5] L S Srinath, Advanced Mechanics of Solids, Tata McGraw-Hill, Third Edition, 2009

[6] S S Rao, The Finite Element Method in Engineering, Butterworth-Heinemann, Fifth Edition, 2010

[7] (2010) The ANSYS website. [Online]. Available: http://www.ansys.com/

[8] Yi-Je Lim, Suvranu De, "Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres," Comput. Methods Appl. Mech. Engrg. 196 (2007) 3011–3024

[9] (2010) The MATLAB website. [Online]. Available: http://www.mathworks.com/

[10] (2010) GPUmat: GPU toolbox for MATLAB. [Online]. Available: http://gp-you.org/