Force controlled elastography with DaVinci Toolkit

Xingtong Liu

Mentors: Preetham Chalasani, Dr. Emad Boctor and Dr. Russell Taylor

May 16th, 2017

Computer Integrated Surgery II

Dr. Russell Taylor

Spring 2017

Johns Hopkins University, Baltimore, MD

Contents

Introduction

Summary

Ultrasound elastography is a good way to detect stiff features at various depths intra-operatively and is not limited to only superficial ones like Palpation. However, free-hand ultrasound elastography is a challenge even for experienced doctors due to various reasons like out of plane motion. Robot assistance has played an important role in surgery due to its high accuracy, safety and efficiency. With proper calibration and force feedback, the robot arm can achieve the most suitable motion for generating elastography images. Therefore, in order to obtain consecutive high-quality elastography images in real time, it is necessary to reach out for the help of a robot.

Goal

da Vinci Research Kit (dVRK) is an open-source platform for researchers to develop algorithms for computer assisted surgery. Previously there was no ultrasound elastography module integrated. In this project, we aim to develop and integrate this module to the dVRK system to allow for relevant future research such as intraoperative registration, tumor recognition and so on. After this project, the dVRK system should be able to receive Ultrasound data from ultrasound machine and display it and be able to compute elastogram with radio-frequency (RF) data.

Proposed solution

In this project, we use ultrasound data acquisition module to acquire data from ultrasound machine and then compute elastogram with received data in ultrasound elastography module. These two modules are developed based on Robot Operating System (ROS) so that they can be easily integrated to the current dVRK system. All tasks are processed in parallel to make it suitable for real-time application.

Background

Ultrasound Imaging

Ultrasonography uses a probe containing multiple acoustic transducer to send pulses of sound into a material. Whenever a sound wave encounters a material with a different density (acoustical impedance), part of the sound wave is reflected back to the probe and is detected as an echo. The time it takes for the echo to travel back to the probe is measured and used to calculate the depth of the tissue interface causing the echo. The greater the difference between acoustic impedances, the larger the echo is.

A-mode (amplitude mode) is the simplest type of ultrasound. A single transducer scans a line through the body with the echoes plotted on screen as a function of depth. In B-mode (brightness mode) ultrasound, a linear array of transducers simultaneously scans a plane through the body that can be viewed as a two-dimensional image on screen. [1]

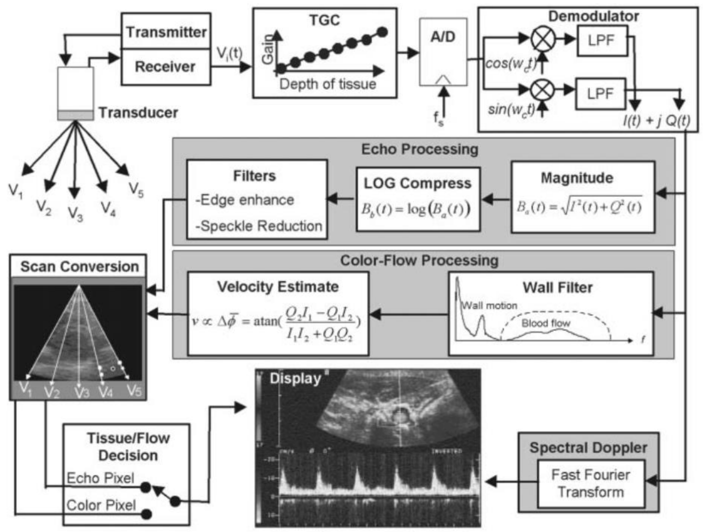

Figure 1 Block diagram of a typical diagnostic ultrasound machine

Ultrasound Probe Calibration

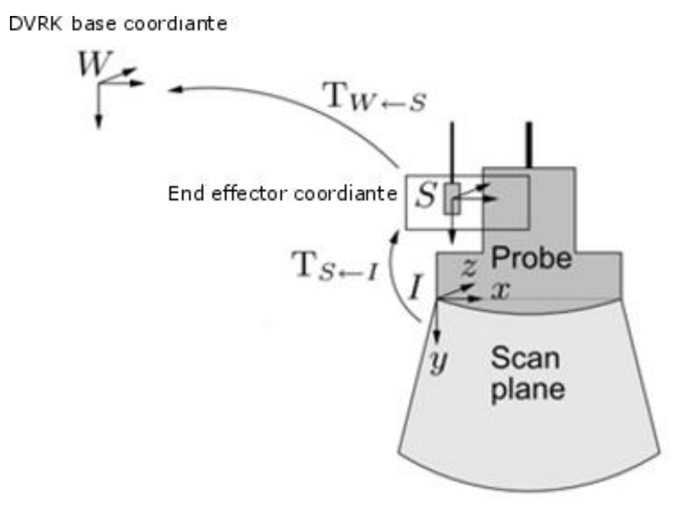

Figure 2 relevant coordinates for Ultrasound probe calibration

The goal for ultrasound probe calibration is to know the transform $T_{S,I}$, which is the transform between the scan plane coordinate I and the coordinate of the transducer on the probe. Also, when the pixel scale factor T_S is unknown, we also need to determine it at the same time during the calibration procedure.

There are two kinds of basic formula for scanning plane calibration. The first one is used when the scale factor is not known and the second one is used when it is known. For the first formula, there are no closed form solution and there are 8 unknowns in total, two scale factors and six calibration parameters. So it needs to be optimized with iterative optimization algorithm like Levenberg-Marquardt algorithm. And in the second formula, there are 6 unknowns, which can be derived with singular value decomposition based closed-form solution proposed by Umeyama. In these formulas, p^W is the position of a fixed point on the calibration phantom with respect to the dVRK base coordinate. $p^{I'_i}$ is the location of the fixed point in corresponding B-mode image. [2]

$$f_{\text{point2}} = \sum_{i=1}^{N} \left| p^W - \mathrm{T}_{W \leftarrow S_i} \mathrm{T}_{S \leftarrow I} \mathrm{T}_s p^{I'_i} \right|$$

$$f_{\text{point3}} = \sum_{i=1}^{N} \left| \mathrm{T}_{W \leftarrow S_i}^{-1} p^W - \mathrm{T}_{S \leftarrow I} \mathrm{T}_s p^{I'_i} \right|$$

Hybrid Force Feedback Control

To generate pre-compressed and post-compressed RF data and calculate elastography quantitatively. It is very important to control the contact force between the elasticity phantom and ultrasound probe to the desired value. There are two kinds of hybrid force feedback control laws that can help us achieve this goal.

Primitive I

$$\delta v = (I - M)\dot{\delta x} + M.\delta v_f$$

Primitive II

$$\begin{aligned} \delta v &= \dot{\delta x}^* + M\delta v_f, \\ &= \delta x + \hat{n} z_a Sin(2\pi\omega t) + M\delta v_f \end{aligned}$$

Where,

$$\delta v_f = \mathcal{G}(F_c - F_d)$$

δx is a constant incremental step along the surface of the model and δv_f is the incremental velocity calculated based on contact force F_c and some desired bias force F_d. M is a projection matrix decomposing the forces in the normal direction of the surface. [3]

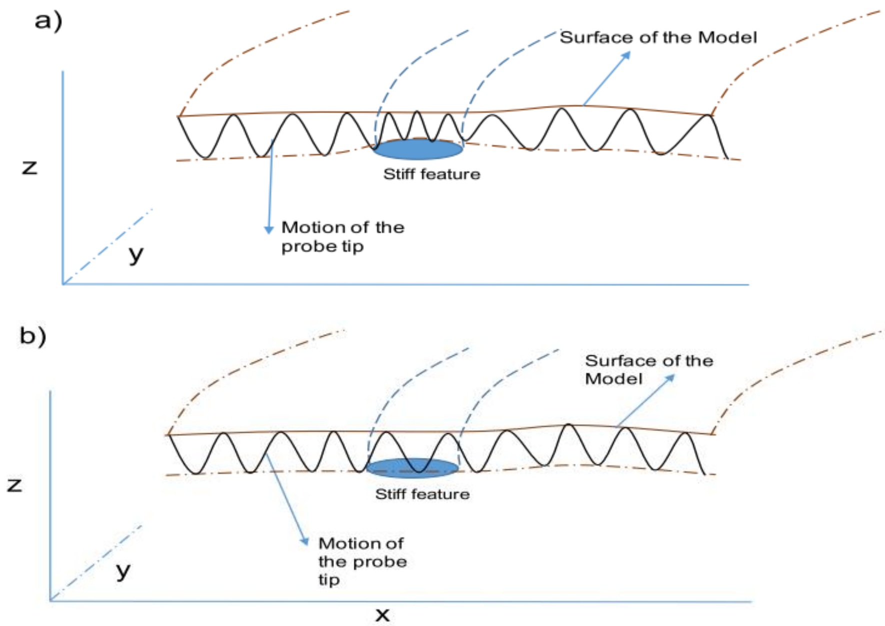

Figure 3 Trajectories of end effector under different hybrid force feedback control law ((a) is Primitive I, (b) is Primitive II)

Elastography

Elastography is an emerging medical imaging method with medical applications such as tumor detection, which computes the spatial variation of the elastic modulus of tissue. There are two types of method, which are Quasi-static method and Dynamic method.

In Quasi-static method, constant stress is applied to the tissue. The elasticity can be calculated based on Hooke's law $\delta = E\varepsilon$. This type of method is easy to implement. But if the exact stress distribution inside the tissue is not known, it is not possible to have a precise estimation of the local Young's modulus.

In Dynamic method, a time-varying force is applied to the tissue -- short transient mechanical force or an oscillatory force with a fixed frequency. Based on shear waves propagation theory $E = 3\rho V_s^2$, it can produce quantitative and higher resolution Young's modulus map. However, this will require a complex system which can generate the shear wave and able to image the small displacements induced by the shear wave. [4]

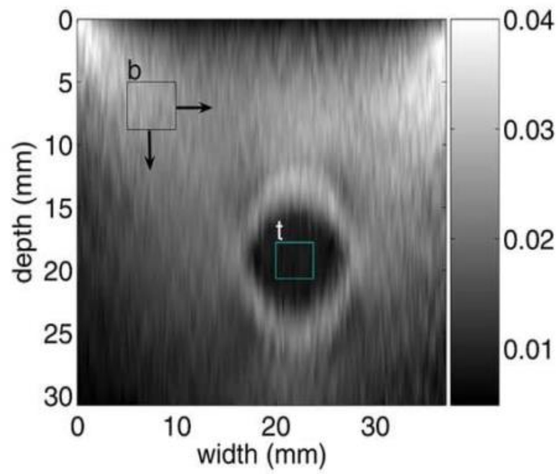

Figure 4 Ultrasound Elastography

Technical Approach

Structure Diagram

The main hardware we use is ultrasound machine, ultrasound probe which is compatible with DaVinci, dVRK hardware, force sensor, elasticity phantom and computer. Robot controlled probe sweeps on the elasticity phantom. The force sensor transmits contact force value to dVRK slave side and then the value is further transmitted to computer. In the current setting, there's no place to attach the force sensor to the ultrasound probe, so it is placed underneath the elasticity phantom. The ultrasound signal is transmitted from the probe to the ultrasound machine. After processing, the beamformed Radio frequency (RF) data is transmitted to the computer. The computer will adapt the control command based on received force value so that the ultrasound probe can move in the desired pattern. Combining force value and corresponding pre-compressed and post-compressed RF data, we can then generate elastogram.

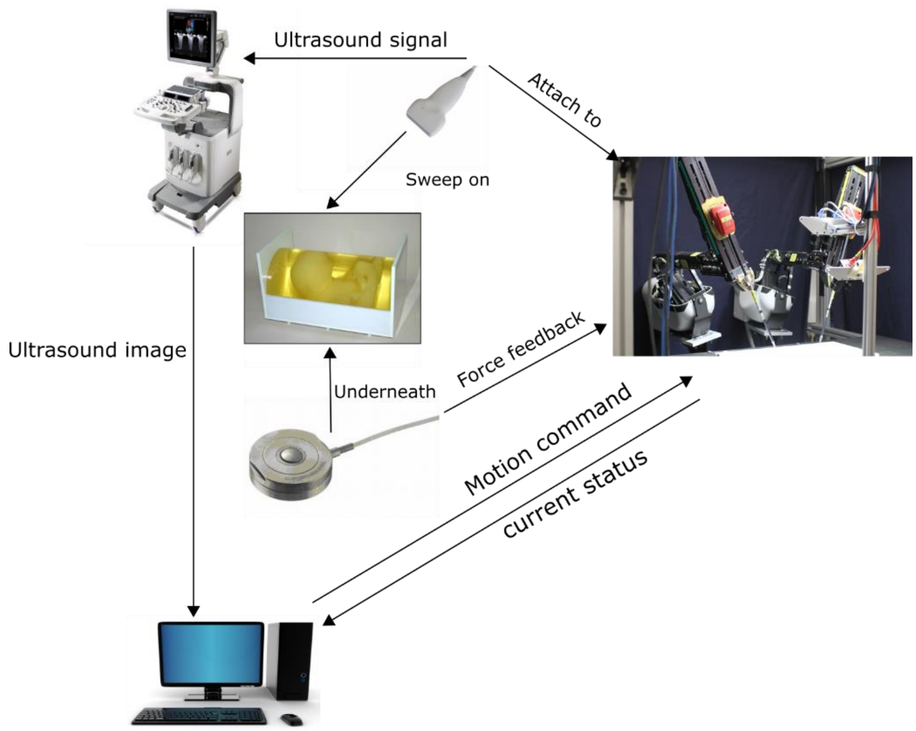

Figure 5 Hardware structure diagram

In this project, there are two new modules integrated into the dVRK software system, which are Ultrasound imaging module and Elastography generating module. Ultrasound imaging module is used to communicate with the ultrasound machine so that dVRK can get access to the RF data. The Elastography generating module is to generate ultrasound elastogram based on force value and RF data.

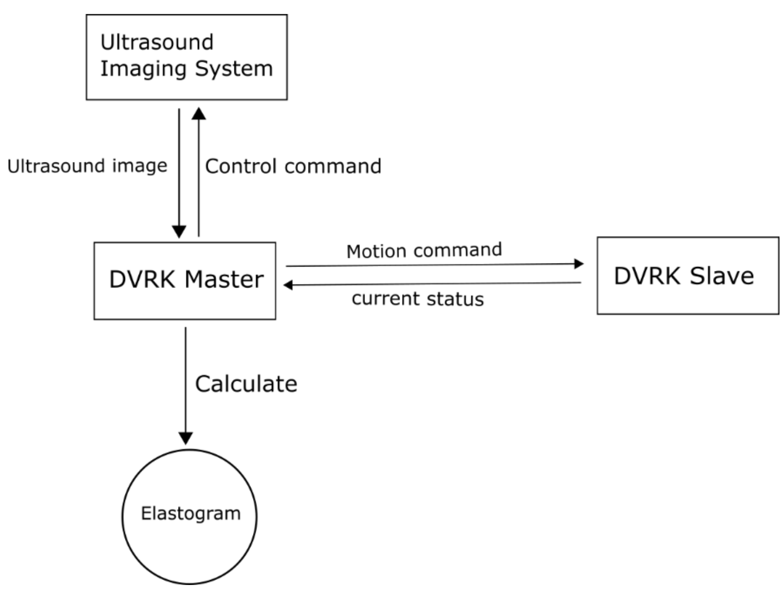

Figure 6 Software structure diagram

Ultrasound Data Acquisition

The communication protocol library we use is OpenIGTLink. OpenIGTLink provides a standardized mechanism for communication among computers and devices in operating rooms for wide variety of image-guided therapy (IGT) applications.

Message Structure

The basic structure for message is as follow. It contains header and body parts.

```
Bytes
0         58            72
+----------+------------+-------------------+-----------+
|  HEADER  | EXT_HEADER |      CONTENT      | META_DATA |
+----------+------------+-------------------+-----------+
           |<---------------- Body -------------------->|
```

The detailed structure of header part is as follow.

```
Bytes
0   2                      14                                      34
+---+----------------------+-------------------------------------+
| V | TYPE                 | DEVICE_NAME                         |
+---+----------------------+-------------------------------------+

34             42             50             58
+--------------+--------------+--------------+
| TIME_STAMP   | BODY_SIZE    | CRC64        |
+--------------+--------------+--------------+
```

State Machine Design

Because there is no control in what can be transmitted through socket between server and client, we should have no assumption about where the start position of the transmitted message is. Therefore, efficiently searching for the valid message in a long bit stream is necessary.

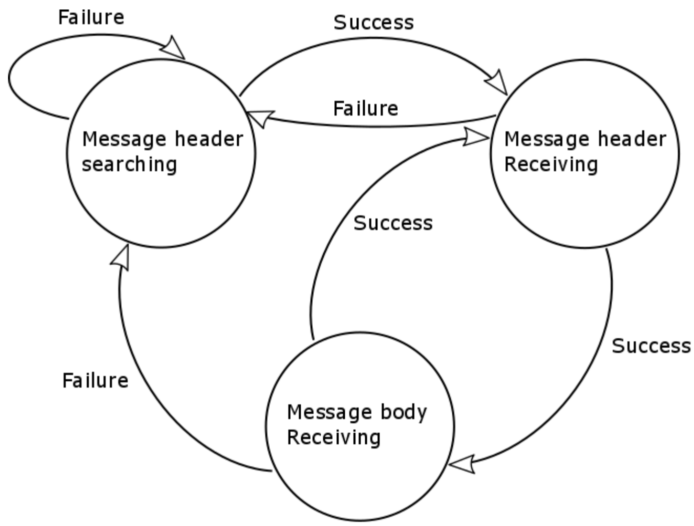

Figure 7 State machine for data acquisition

The above figure is the state machine used for data acquisition. To efficiently search valid message in the bit stream, there is a state called Message header searching. I use first 14 bytes to locate the desired type of valid message we want. The first two bytes called V, the only possible value for V is 1 or 2. The following character array represents the type of the message. In this application, we want to receive the RF data or B-Mode image. So the value for this array should be IMAGE. With 14 bytes being correct, there is basically no chance (2^{-111}) that this is not a valid message header. So the next step should be receiving the entire header, which is state Message header receiving. If no valid header is found, we should continue searching the valid subset of header in the bit stream.

In the Message header receiving state, we need to receive the entire message header to allocate memory based on information in header such as DEVICE_NAME and BODY_SIZE. There are two types of DEVICE_NAME we expect to receive. One is "MD_US_2D" and another one is "MD_BMODE_2D", which corresponds to RF data and B-Mode image. BODY_SIZE is the size of message body and this is used to dynamically allocate memory for message. CRC64 in message header is to ensure that this is a valid one, which provides further validation check. If it is corrupted or invalid, return back to Message header searching state.

Message body receiving part is to receive the message body, which, in this application, is image. After pushing image into queue for other threads to process. The state returns back to Message header receiving. The rationale is that usually the messages are transmitted consecutively. So after receiving the message body for previous message, we expect to receive the message header of a new message.

Multiple Threads and Socket I/O Model

For real-time application, the program usually needs to process multiple tasks simultaneously. Therefore, it is necessary to use multi-thread programming. Because ultrasound data acquisition is a module which runs all the time, we will assign one thread to it. For elastogram computation, sometimes some frames cannot be processed in time, so we need a way to store these frames to be processed. Concurrent queue is used in this project to store incoming frames of RF data. It is multi-thread safe and the data in queue can be used in multiple threads. The way to use queue can be demonstrated as the follow figure.

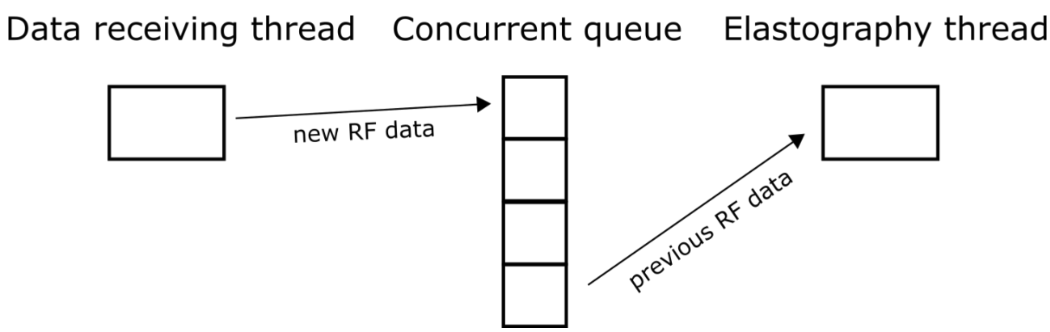

Figure 8 Data communication between threads

For socket programming, the traditional model is synchronous blocking I/O model. In that model, if we need to read data from a socket, we need to wait without doing anything else until all data has been read. Also, because it is synchronous, we need a loop to constantly check whether there is data ready to be read, which will consume a lot of CPU resources. Overall, this I/O model is not efficient.

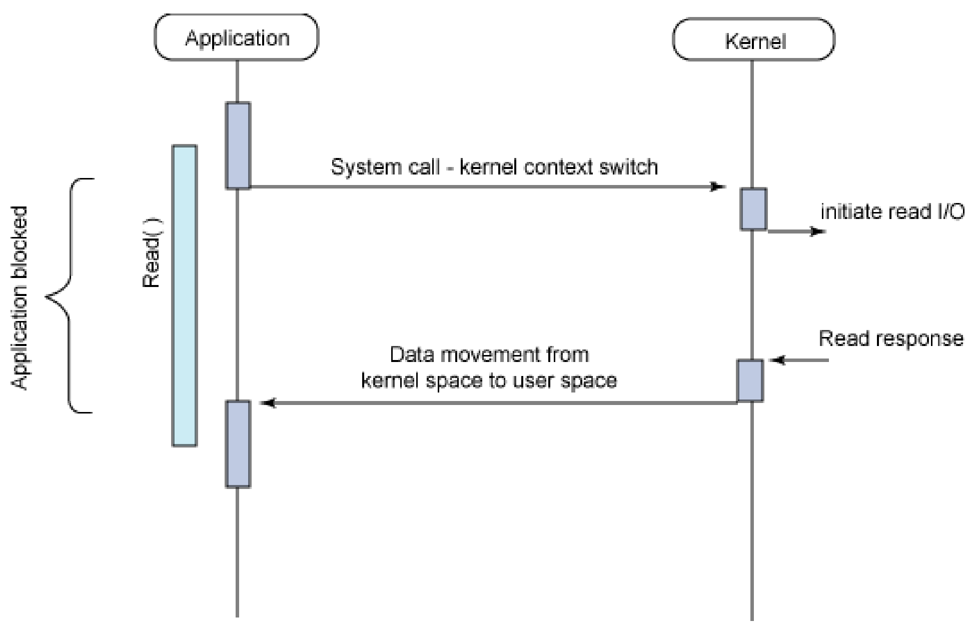

Figure 9 Typical flow of synchronous blocking I/O model

Asynchronous non-blocking I/O model is a better way. In this way, only when the data is transmitted completely, the kernel will emit a signal to let a thread know so that it can move data from kernel space to user space. Before the data is ready, there is no blocking in user space, therefore other processing can be done in the meantime. The library we use for this model is Libevent.

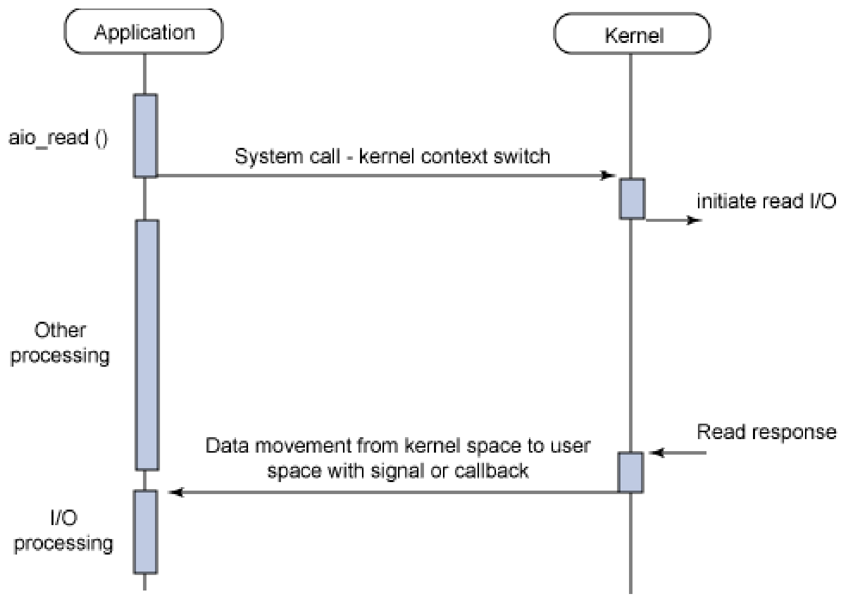

Figure 10 Typical flow of asynchronous non-blocking I/O model

Ultrasound Elastography

Current elastography method is based on cross-correlation. The mechanism for this method is to compare two segments from pre-compressed and post-compressed RF signals in order to find the local time shift between these two signals. We can demonstrate it better by showing two spring examples.

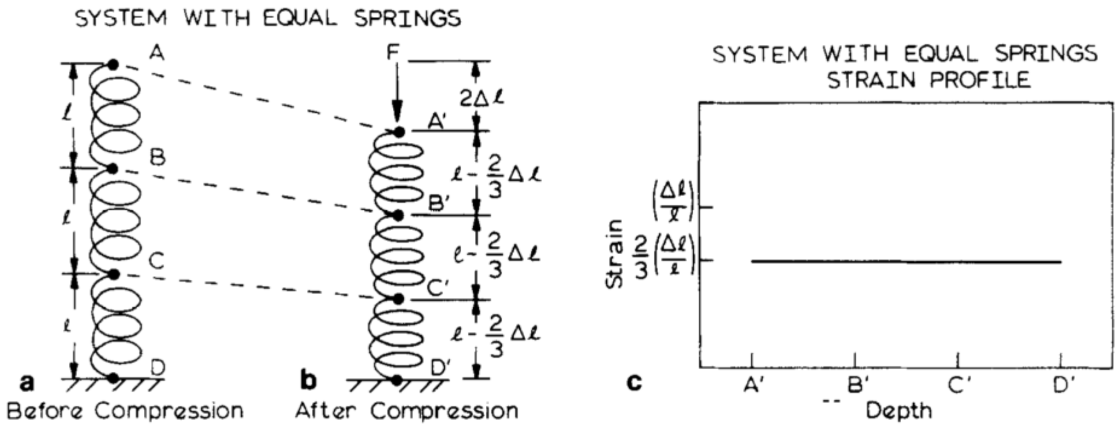

Figure 11 Measurement of strain in a one dimensional cascaded spring system of equal spring constants. (a) Pre-compression state; (b) post-compression state; (c) Strain profile. Note strain constancy.

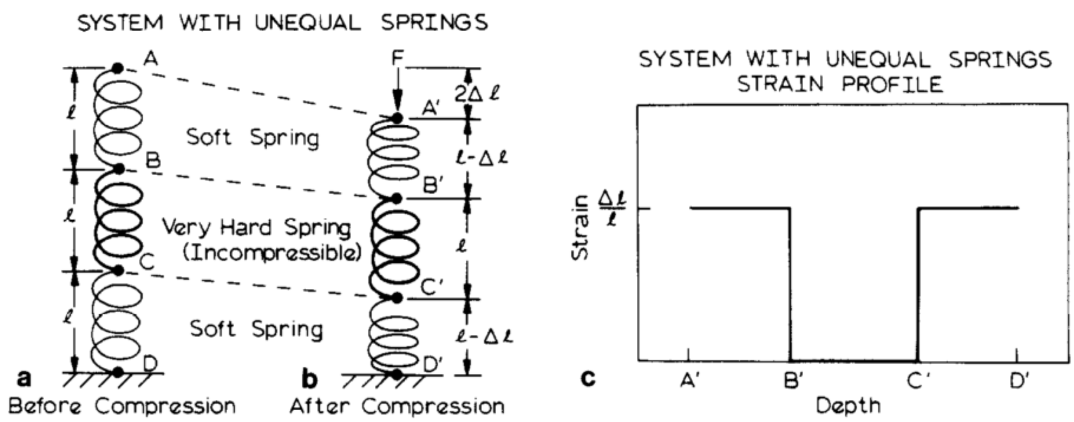

Figure 12 Measurement of strain in a one dimensional cascaded spring system of unequal spring constants. (a) Pre-compression states; (b) post-compression state; (c) Strain profile. Observe that the strain in the softer springs depends on the presence of the stiff spring.

To measure the elasticity of each spring, we need to know the strain of them, which means how much they are compressed. For Spring AB. The strain can be calculated as,

$$\frac{|A - A'| - |B - B'|}{dz}$$

where dz is the overall compression of the system. A, A', B, B' are the positions of the corresponding points on spring.

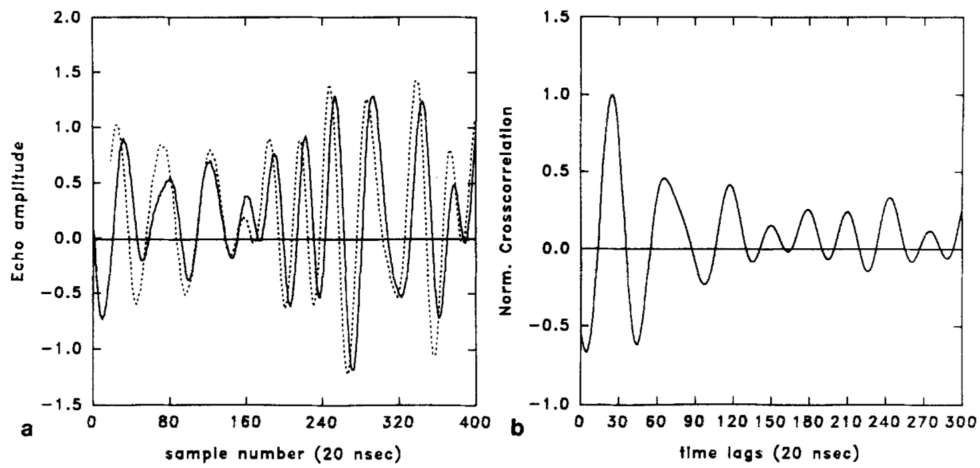

Figure 13 (a) A typical segment pair; (b) resulting cross-correlation function between segments in (a)

This is basically the same for RF data. The only differences are that the displacement of RF data is represented in time shift and the time shift is calculated by computing the cross correlation between two corresponding segments. In other words, the tissue strain is typically estimated from the gradient of tissue displacements. The local tissue displacements are estimated from the time delays of gated pre- and post-compression echo signal. Time delays are generally estimated from the location of the peak of the cross-correlation function between the gated pre- and post-compression.

The strain profile is defined by

$$s_i = \frac{t_{i+1} - t_i}{2dz/c}$$

where t_i is the time shift for ith segment pair, dz is the overall compression of the system and c is the speed of sound inside the tissue.

If we know the stress distribution ε inside the tissue, then based on Hooke's law $\delta = E\varepsilon$, where δ is the strain, we can estimate elasticity E of the tissue.[5]

Cross-correlation method has both pros and cons. It is simple and easy to implement. Also it can be paralleled and has GPU implementation so this method can be used in real-time application. However, this method is very sensitive to noise and signal decorrelation. But for small strain, global stretching can almost compensate for the signal decorrelation. This type of method use gradient operation and it is well-known that gradient operation amplifies noise in the displacement estimates. There are also some new methods such as adaptive stretching to deal with this issue but in this project we only use the basic cross-correlation method. Further improvement is needed for future usage.

Results

Figure 14 Phantom for verification

The above phantom is used for testing. There are two stiff spheres in gel. The corresponding B-Mode image and displacement image are as follows. As you can see, there are two white curves in B-Mode image. I think the reason why there are no spheres is because the echo signals are all reflected back to the transducers when they hit the surface of these two stiff spheres, which should be two curves. From the corresponding displacement image, from where the elasticity modulus can be derived directly if the stress distribution is known, we can also find similar patterns.

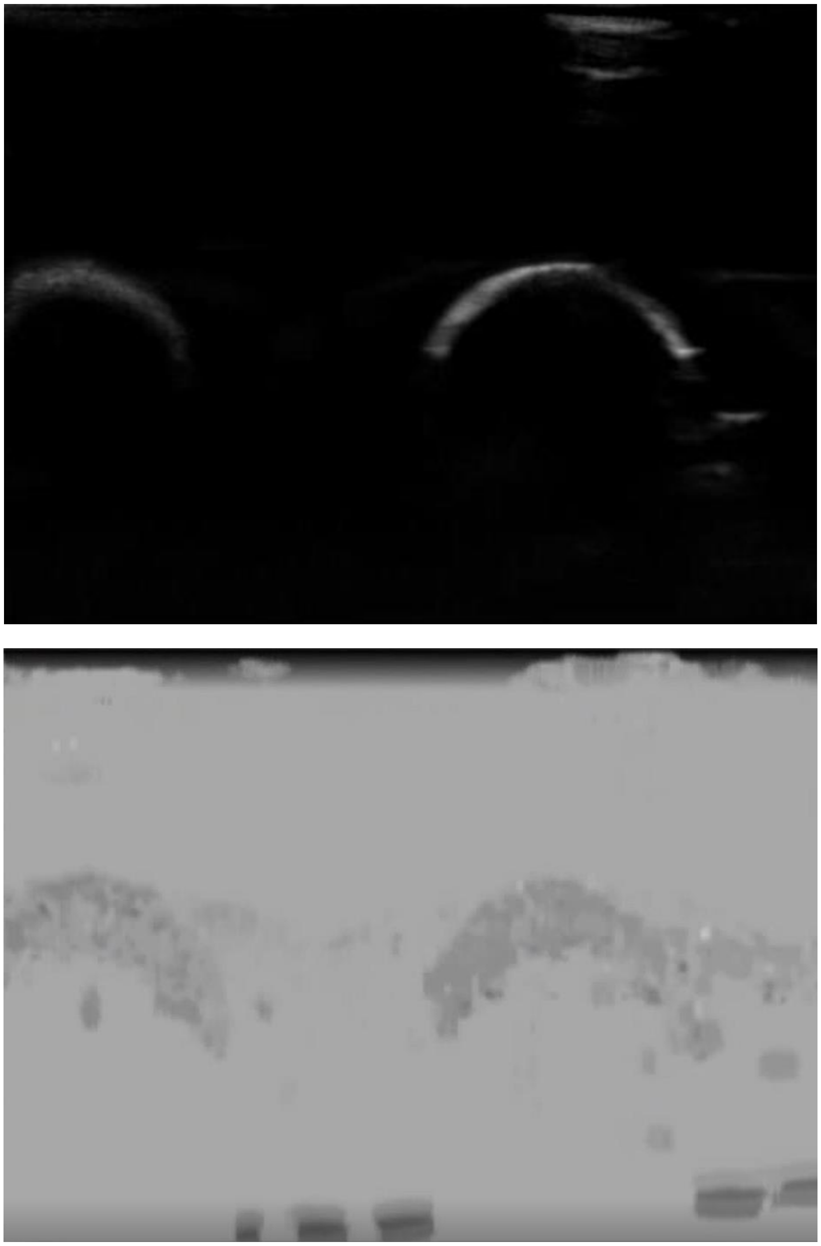

Figure 15 B-Mode image and corresponding displacement map

Challenges and Issues

1. The mechanical issue of current da Vinci ultrasound probe makes scan plane calibration impossible. Also, because the current probe can be easily passively rotated when being pressed against the phantom, it is not suitable for teleoperating palpation. We need to replace it with a new drop-in probe.
2. Current cross-correlation elastography method is sensitive to signal decorrelation. We need to further improve the algorithm with technique advancements such as global stretching or adaptive stretching to make it more robust and useful in practice.

Significance

We have developed a software which can acquire ultrasound data and compute ultrasound elastography in real time. This can be integrated with dVRK or any other Robot Operating System (ROS)-based system. Though currently the DaVinCi ultrasound probe has some mechanical issues and the elastography method needs to be further improved with advanced techniques, which prevent it from being practical for now in dVRK system, with drop-in ultrasound probe and advancement in algorithm, the software will enable many researches based on ultrasound elastography with dVRK system.

Management Summary

Deliverables

Minimum:

DaVinci ultrasound tool and ultrasound system integration into dVRK system -- Complete

Completed kinematic configuration for ultrasound probe so that the probe is able to be controlled in teleoperation mode. Completed ultrasound imaging system integration. The ultrasound image can now be received and displayed in dVRK system.

Expected:

Ultrasound imaging with force feedback control – Complete

Because the force feedback control module has been implemented before, this deliverable just includes verifying whether the ultrasound imaging module can work well together with force feedback control module. We have verified that the ultrasound imaging module can work well with dVRK system.

Maximum:

Ultrasound elastography with dVRK – Partially complete

Completed Ultrasound elastography module. We have verified that this module works well with dVRK system. Because of the mechanical issue of the current ultrasound probe, the scan plane calibration cannot be completed. Also, because the transducer part of the probe can be easily passively rotated, compressing phantom with that probe by teleoperation seems not feasible either.

Acknowledgements

This project is supported by Alexis Cheng, Preetham Chalasani, Dr. Emad Boctor and Dr. Russell Taylor. I would like to thank Alexis and Dr. Boctor for supporting the project with ultrasound machine, ultrasound probe and elasticity phantom and guidance. Also, I would like to thank Preetham for helping me with the experiments and thank Dr. Taylor for valuable general guidance.

References

[1] G. York and Y. Kim, “Ultrasound processing and computing: review and future directions.,” *Annu. Rev. Biomed. Eng.*, vol. 1, pp. 559–588, 1999.

[2] P. Hsu, R. W. Prager, P. Hsu, R. W. Prager, A. H. Gee, and G. M. Treece, “Freehand 3-D Ultrasound Calibration: A Review,” *Engineering*, no. December, 2007.

[3] P. Chalasani, L. Wang, R. Roy, N. Simaan, R. H. Taylor, and M. Kobilarov, “Concurrent Nonparametric Estimation of Organ Geometry and Tissue Stiffness Using Continuous Adaptive Palpation.”

[4] J. L. Gennisson, T. Deffieux, M. Fink, and M. Tanter, “Ultrasound elastography: Principles and techniques,” *Diagn. Interv. Imaging*, vol. 94, no. 5, pp. 487–495, 2013.

[5] J. Ophir, I. C??spedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: A quantitative method for imaging the elasticity of biological tissues,” *Ultrason. Imaging*, vol. 13, no. 2, pp. 111–134, 1991.