Winter 12-15-2014

Real-time Temperature Imaging Using Ultrasonic Change in Backscattered Energy

Weiyuan Zhao
Washington University in St Louis

Washington University in St. Louis

School of Engineering and Applied Science

Department of Electrical and Systems Engineering

Thesis Examination Committee:
R. Martin Arthur
William L. Straube
Jason W. Trobaugh

REAL-TIME TEMPERATURE IMAGING USING ULTRASONIC CHANGE IN BACKSCATTERED ENERGY

by

Weiyuan Zhao

A thesis presented to the Graduate School of Arts and Sciences
of Washington University in partial fulfillment of the
requirements for the degree of

Master of Science

December 2014
Saint Louis, Missouri

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my advisor Dr. R. Martin Arthur for his enormous encouragement and support over the past two years. It's a great honor and my good fortune to have him as my advisor. I learned so much from him not only in the academic field, but more importantly about how to live a life with meaning. I can't describe how much I was influenced by him through our discussions about music, culture and connecting the dots. All the knowledge I gained from him will be the priceless treasure for the rest of my life.

I want to express my sincere gratitude towards Dr. Jason W. Trobaugh for his valuable inputs to my thesis and suggestions to our future work. Also to Professor William L. Straube from the Department of Radiation Oncology in Washington University Medical School for his help and support to our experimental setup.

I feel very proud and lucky to be able to study in the Department of Electrical and Systems Engineering. A big thank you to the faculty, staff and classmates for their consistent and passionate help.

Finally, I would like to dedicate a special appreciation to my friends, co-workers and family for their unconditional love and constant prayers. I wouldn't be able to complete this journey without their support.

Weiyuan Zhao

Washington University in Saint Louis
December 2014

Dedicated to my grandparents, my aunt & uncle, and my parents.

Dedicated to everyone who helped and supported me.

ABSTRACT OF THE THESIS

REAL-TIME TEMPERATURE IMAGING USING ULTRASONIC CHANGE IN BACKSCATTERED ENERGY

by

Weiyuan Zhao

Master of Science in Electrical and Systems Engineering

Washington University in St. Louis, December 2014

Research Advisor: Professor R. Martin Arthur

Thermal therapy from low-temperature cryosurgery to high-temperature ablation of tumors and unwanted electrical pathways has gained increased attention. Temperature imaging (TI) from magnetic resonance (MR) studies is the de facto standard for volumetric estimation of temperature. The high cost and the difficulty in cooperating with heating instruments of MR systems have limited its wider implementation. Ultrasound on the other hand, has the advantages of being cheap, portable, non-invasive and non-ionizing. Ultrasonic properties that change with temperature include speed of sound, acoustic attenuation coefficient, and change in backscattered energy (CBE). Our group predicted for single scatterers and simulated for scatterer populations that CBE would change monotonically with temperature. We also estimated temperature using CBE in 1D, 2D, and 3D in different tissue types with 1^{o}C accuracy. An obstacle to clinical application of CBE TI is estimating temperature in real time, which is limited by time for motion compensation (MC).

To achieve real-time TI, we implemented a two-computer architecture. Our Terason 3000 ultrasonic imaging system collected and sent raw images over a jtcp connection to a computer, which estimated temperatures images. The TI machine was an HP Envy Phoenix 810 with a GeForce GTX 770 GPU card. The TI computer performed motion compensation and extracted temperature images. Turkey specimens were imaged during heating with hot water (75^oC) in 1 cm tube. Total heating time was 1200 sec, with a 30 sec interval between image acquisitions, tissue temperature was monitored with thermocouples.

Over six experiments at 3 thermocouple sites, the accuracy of CBE TI was 0.8 ± 0.7^oC. Using its CPU, the TI computer updated temperature images using rigid MC in 4 sec, and using more nearly accurate nonrigid MC in 7 sec. Nonrigid MC time was reduced to 0.2 sec using the GPU processor along with optimization of the MC algorithm. Calculation of CBE in MC images and conversion of CBE to TI takes less than an additional 0.1 sec.

With TI time reduced to < 0.3 sec, the limit to real-time CBE TI now lies with the Terason 3000 system. It takes about 5 sec to transform an ultrasonic image in its native format to Matlab before sending it to the TI computer. Therefore, we believe CBE TI can be done at a 1 Hz frame rate with $< 1^o$C error if conversion to Matlab in the Terason 3000 can be reduced to less than 0.7 sec.

Chapter 1

Introduction: Thermal Therapy

Thermal therapy is an adjuvant to radiation and chemotherapy in the treatment of cancer [27]. In addition, thermal therapy can be used to ablate aberrant electrical pathways in the heart. Thermal therapies range from low-temperature cryosurgery to hyperthermia and high temperature ablation of tumors has gained vast attention in recent years [2, 18]. Because thermal therapy may have fewer toxic side effects or or less unwanted tissue damage compared to radiation or chemotherapy, it may be an appealing adjuvant or even a stand alone therapy.

Thermal therapy with temperature in the range of 41 ^{o}C to 45 ^{o}C is also called hyperthermia, which has been employed in cancer treatment [14]. Hyperthermia will damage or kill the cancer cells. It can also make the cancer cells more sensitive to the effects of radiation and some anti-cancer medication, thus it would significantly enhance the effectiveness of radiotherapy and chemotherapy [19]. Very high temperatures, above 50 ^{o}C (122 ^{o}F), are used for ablation (direct destruction) of some tumors or unwanted electrical pathways [29].

The control of thermal dosage is the key to safely perform thermal therapy [27]. Because there is a possibility of damaging adjacent healthy tissue if there is not enough information about how the temperature spreads during heating. This limitation has been a major impediment to the wider implementation of thermal therapy. Thus, there is a need for accurately monitoring the temperature in body tissue during thermal therapy.

1.1 Temperature Monitoring

Many techniques have been investigated for the use of monitoring tissue temperature during thermal therapy. The techniques roughly fall into two categories: 1) Sparse and 2) Volumetric.

1.1.1 Temperature Probes

Such thermometry includes the using of thermocouples and fiber optics as thermometers [23]. Both approaches in temperature measurement, however, are invasive and sparse, which limits the temperature estimation accuracy of the surrounding tissue that's not monitored by probes. Hence it cannot provide comprehensive temperature information, nor can it guide the thermal therapy effectively. In addition, these methods are all invasive, which will cause extra risk to the patient. Thus methods that are noninvasive and volumetric are needed.

1.1.2 Volumetric Imaging

A number of volumetric imaging models are being investigated, including electrical impedance tomography (EIT), microwave, magnetic resonance imaging (MRI) and ultrasound [7, 26]. Among which, EIT methods do not hold enough accuracy in the hyperthermia temperature range [35]. Furthermore, microwave radiometry type of methods do not have good enough spatial resolution when applied to deep targets [22].

1.2 MRI Temperature Imaging

MRI exploits the shift in the proton resonance frequency with temperature [17, 26]. MRI is considered the *de facto* standards to measure temperature non-invasively and volumetrically. It has shown satisfactory accuracy and spacial resolution with 1^oC accuracy. The primary disadvantages of MRI, however, are that it is expensive, it requires huge facilities and MRI-compatible heating devices [10]. Ultrasound, on the other hand, has the advantages of being

cheap, portable and non-ironizing, and would be a better modality to be widely used in clinical environment.

1.3 Ultrasonic Temperature Imaging

Ultrasound as a non-invasive thermometer exploits the following three tissue properties of ultraound: 1) acoustic attenuation coefficient [6], 2) echo shifts caused by thermal expansion and changes in speed of sound (SOS) [1], and 3) change in backscattered energy (CBE) [2]. Previous research has shown that attenuation does not change significantly in the hyperthermia temperature range, and echo-shift based modality requires prior knowledge of the temperature dependence on the change in SOS and tissue thermal expansion [2].

Our group has been investigating the use of CBE for temperature monitoring. Straube and Arthur had predicted a monotonic change in CBE with rise in temperature using a theoretical model with containing lipid and aqueous subwavelength scatterers in an aqueous medium [28, 6]. Simulation studies scatterer populations have also shown a similar monotonic trend in change in backscatter energy with temperature [32]. We also estimated temperature using CBE in 1D, 2D, and 3D in different tissue types with 1^oC accuracy [5, 3].

1.4 Objective

The previous work of our group has proved that change in backscattered energy can be used for accurate thermometry. CBE provides a cheap and portable modality to the current *de facto* standard MRI. Our group has carried out studies of CBE temperature estimation both *in vitro*, and *in vivo* system [7]. *In vitro* non-uniform heating studies showed that CBE temperature estimation was accurate to within about 1^oC over 7x7 mm regions [10].

In this study, our objective was to apply non-invasive CBE temperature imaging (TI) during non-uniform heating in real time. The primary barrier to real-time CBE TI is the time required to compensate for motion during heating. Motion compensation is needed for accurate

calculation of CBE from which TIs can be inferred. Uncompensated motion masquerades as spurious CBE.

To achieve real-time TI: 1) We tested a two-computer architecture in which images from our Terason 3000 ultrasonic imaging system were sent via internet communication to a dedicated machine to perform motion compensation, calculate CBE and extract temperature images, 2) We tested CBE TI in turkey specimens during local heating with hot water (75^oC) in 1 cm tube, and 3) We tested for reduction in motion compensation calculation time with improvements in the motion compensation algorithm and the use of a GPU array for finding motion in the TI machine. Our goal was to reduce computation time for 2D TI in a 1871x128 (pixels) area to less than 1 sec.

Chapter 2

Ultrasonic Temperature Imaging

The implementation of thermal therapy in cancer treatment and other clinical application is getting more and more attention in recent years. There has been a growing need for a non-invasive temperature monitoring modality to effectively guide the thermal treatment process [1]. A clinically useful method requires accurately (0.5^oC) measure 3D temperature distributions within 1 cm^3 volumes [5]. From Chapter 1, we learned that among all the other non-invasive thermometry methods, ultrasound stands out as a portable, cheap and non-ironizing technique [34]. Because of its promising properties, the use of ultrasound in the temperature imaging area has been largely studied over the recent decades.

2.1 Ultrasound

In addition to being convenient, non-invasive and non-expensive, ultrasound also requires simple signal processing [34, 19]. These attributes make it an attractive method to use for temperature estimation if an ultrasonic parameter, which is dependent on temperature, can be found, measured, and calibrated. The ultrasonic parameters examined for their dependence on temperature are acoustic attenuation coefficient, speed of sound and ultrasonic backscatter energy [28].

2.1.1 Attenuation

The effects of temperature on ultrasonic attenuation and absorption have been explored by other groups. They found that attenuation was highly dependent on temperature, but only at temperatures $> 50^oC$. The attenuation coefficient change is small in the hyperthermia range (41^oC to 45^oC) [36]. It is discovered that the change in attenuation also varied for different tissue types and time of heating [33]. Thus attenuation is of interest for thermometry at temperatures above 50^oC. We need to, however, find a parameter that is highly dependent on temperature in the clinical hyperthermia range.

2.1.2 Speed of Sound and Echo Shift

Speed of sound (SOS) was the first ultrasonic parameter whose temperature dependence was researched [31]. The temperature estimation using SOS requires accurate information of distance traveled by the ultrasound wave and the time it took [31]. Thus it was never applied in clinical environment where the acquisition of the above information would be difficult in *in vivo* system.

Echo shift as another temperature dependent parameter has received more attention in the recent years [21]. Multiple groups have been able to estimate temperature both theoretically and experimentally in phantoms through tracking scattering volumes and measuring the time shift of received echoes [20]. Varghese and coworkers used this displacement gradient (due to SOS changes and thermal expansion) along with accumulated echo shifts to track temperatures in phantoms and estimated temperature within 0.5^oC [31].

Use of echo shift and displacement gradient, however, calls for prior knowledge of SOS and thermal expanse which is hard to obtain *in vivo*. Similar to attenuation-based methods, most of these efforts have been geared toward high intensity focused ultrasound therapy ($> 60^oC$) and may not be suitable for monitoring of moderate-temperature hyperthermia [9].

2.1.3 Back Scattered Energy

To find an ultrasonic parameter that changed monotonically with temperature, we modeled the backscattered energy from individual scatterers to an interrogating ultrasonic wave [28]. The change in backscattered energy due to temperature was primarily dependent on the changes in SOS and density of the medium compared to their values in sub-wavelength inhomogeneities (scatterers) within the medium. Our predicted change in backscattered energy (CBE) at any temperature T with respect to its value at some reference temperature T_R is

$$CBE(T) = \frac{\alpha(T_R)}{\alpha T} \frac{\eta(T)}{\eta(T_R)} \frac{[1 - e^{-2\alpha(T)x}]}{[1 - e^{-2\alpha(T_R)x}]} \tag{2.1}$$

where, $\alpha(T)$ is the attenuation within the tissue volume as a functions of temperature, and $\eta(T)$ is the backscattered coefficient of the tissue volume. Distance x is the path length in the tissue volume. We studied effects of noise level on temperature imaging using Pennes' bioheat equation (Eq. 2.2) [25]. For the in vitro case, in which perfusion and metabolism can be neglected, the heat flow equation at temperature T becomes

$$\rho C_p \frac{\partial T}{\partial t} = \nabla(k \nabla T) + Q \tag{2.2}$$

where ρ is density, C_p is specific heat, k is heat conductivity, and Q is the heat delivered to the specimen. Our group has investigated methods based on change in backscattered energy (CBE) with temperature variation. The backscattered signal depends on attenuation, velocity of sound in the tissue, density, backscatter coefficient of the tissue and on the transducer properties [28].

Analysis of backscattering power variation with temperature has revealed that it primarily depends on the temperature-dependent backscatter coefficient of tissue [28, 19, 1]. Thus, this method exploits the presence of tissue inhomogeneities and does not require prior knowledge of SOS as a function of temperature.

2.2 CBE

It was predicted that change in ultrasonic backscattered energy (CBE) would change monotonically with temperature by Straube and Arthur [28]. Further investigation were carried out to prove this prediction.

2.2.1 Theoretical Model

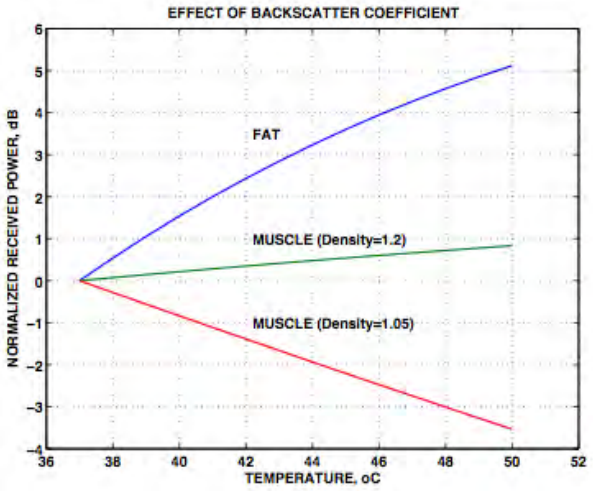


Figure 2.1: Theoretical prediction of CBE using single scatterers in a liver medium. The curve labeled Fat was the result of assuming a lipid-based scatterer in a water-based medium. The curve labeled Muscle was the result of assuming an aqueous scatterer in the same water-based medium used for the lipid scatterer. From Straube *et al* 1994 [28].

Straube and Arthur created a theoretical model for the average backscattered energy from a random distribution of scatterers as shown in Fig. 2.1. In this model, it was shown that the change in backscattered energy could increase or decrease depending on the type of inhomogeneity, for example lipid or aqueous, caused the scattering. Thus, depending on the spatial resolution and the type of scatterers within a given volume, the change in backscattered energy could vary greatly for a given set of scatterers. This model provided a basis for temperature imaging using change in backscattered energy [28].

2.2.2 Simulation of Scatterer Populations

In order to investigate CBE for populations of scatterers, our group has developed an ultrasonic image simulation model [32]. This simulation model included temperature dependence for individual scatterers based on predictions from our theoretical model [28]. This simu-

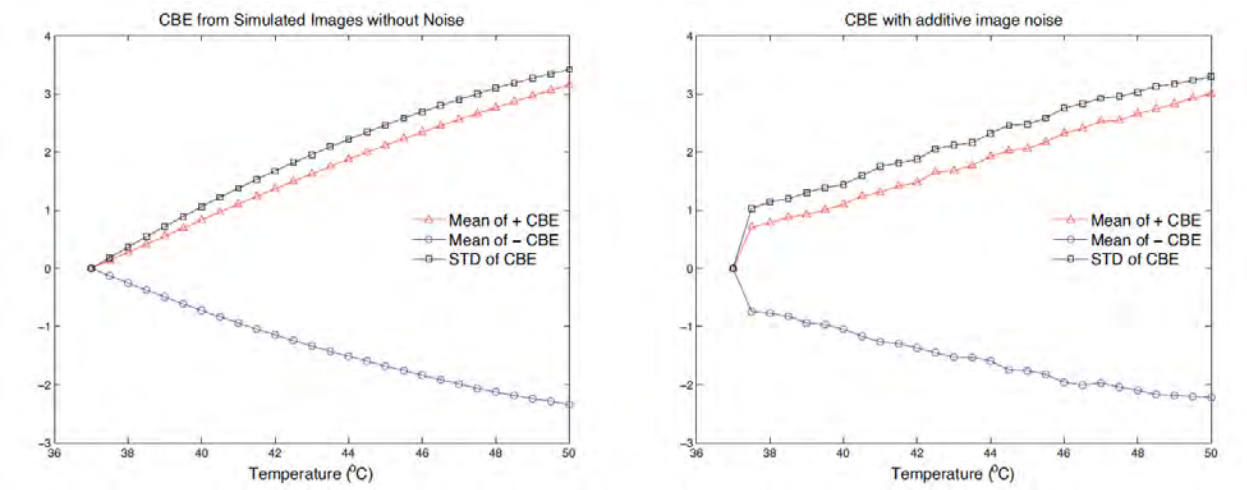


Figure 2.2: Left) CBE for 37^{o}C to 50^{o}C computed from simulated images without additive noise, which agrees with the prediction in Fig. 2.1. Right) CBE for 37^{o}C to 50^{o}C computed from simulated B-mode images with additive noise, which agrees with the experimental results (Fig. 4.36 part (a), for example). From Trobaugh *et al* 2008 [32].

lation model was used to study the effects of signal-to-noise ratio, region size and scatterer population on the CBE measurements. As shown in Fig. 2.2, the simulated CBE without noise (left) is in consistency with our theoretical model Fig. 2.1. The simulated CBE with additive noise is shown in Fig. 2.2 on the right, and it matches our measurement results shown in Fig. 2.3. The temperature dependence of CBE of multiple scatterers using the discrete scatterer model and linear image modeling [32]. The simulation study further confirmed the previous theoretical model with monotonic variation for individual pixel measurements and for image regions [28]. The simulation study also evaluated the tradeoffs between temperature accuracy and spatial resolution, the effects of SNR and scatterer population. Using the simulation study, it was shown that with a 1 x 3 cm region size and 19dB signal to noise ratio, it is possible to estimate temperature within 0.5^{o}C [32].

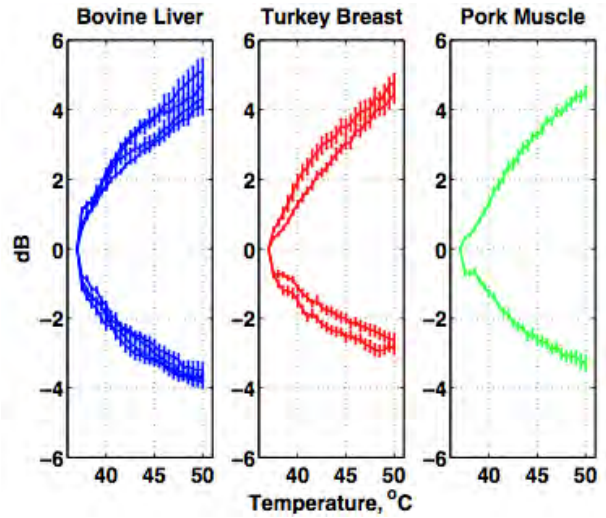


Figure 2.3: Means of measured CBE in positive and negative regions of BE images in four specimens of bovine liver, two of turkey breast, and one of pork muscle . From Arthur and Trobaugh. 2005 [9]

2.2.3 Motion Compensation

For every temperature imaging estimation method, it is important to track and compensate for motion between images [9], so that CBE could be measured accurately. Uncompensated motions introduce spurious CBE [32]. Thus, it is important to correct for the apparent change in backscattered energy due to motion in the region of interest to ensure that CBE is due to thermal effects only. For motion compensation, image features have been tracked using cross-correlation techniques used by investigators using the echo-shift techniques for temperature estimation [5].

There are two primary objectives for motion tracking in ultrasonic temperature imaging. For methods based on echo shifts, tracking is local and most effective in 1D. For methods that use signal strength changes, tracking is global, that is over a tissue volume and in 2D or 3D, where motion is likely to be non-rigid [2]. See Fig.2.4 for a illustrative view of the 3D motion.

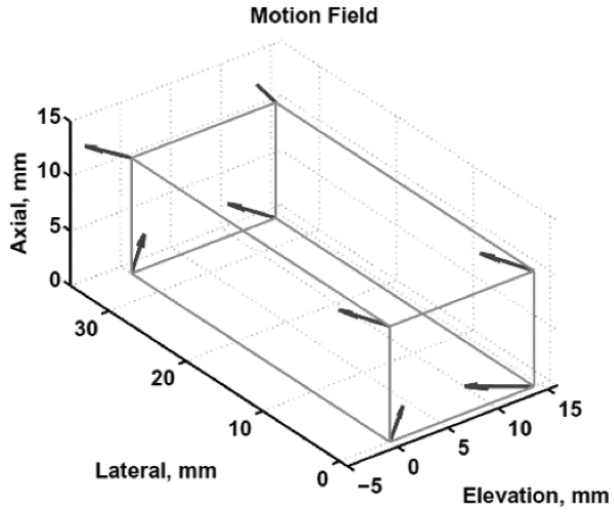


Figure 2.4: Nonrigid motion in 3D. Arrows represent the motion at 8 corners of a 3D tissue volume. The values and directions of the motion may be different at each location in the tissue volume. From Arthur et al. 2010 [3]

2.2.4 Measurements in 1D, 2D and 3D

Experimental results have confirmed the monotonic change in CBE in both 1D ultrasonic signals and in 2D, 3D ultrasound images with *in vitro* specimens, in the temperature range of 37-50^{o}C [5]. We measured CBE values similar to our predictions in bovine liver, turkey breast, and pork muscle in 1D [5]. These measurements were corrected manually for changes in the axial position of echo signals with temperature [9]. To investigate the effect of temperature on changes in backscattered energy in 2-D, we imaged 1-cm thick samples of bovine liver, turkey breast, and pork muscle during heating in a water bath from 37 to 50^{o}C [9] see Fig. 2.3. 3D measurements of CBE was done in turkey breast muscle by moving a phased-array transducer in elevation while maintaining the temperature in the tissue using a thermal controller and thermocouples. Estimated temperatures for 1 cm^3 volumes using the mean value for CBE sensitivity of 0.3dB/^{o}C , with error of 0.3 $\pm$ 0.5^{o}C [10] see Fig. 2.5 and Fig. 2.6.

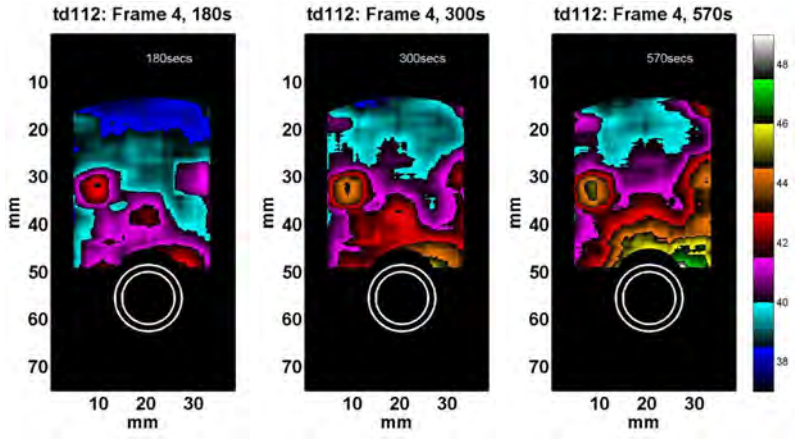


Figure 2.5: Estimated temperature images in turkey breast muscle for the center frame of a 3D dataset from experiment td112. A 7×7 mm window was used to compute CBE maps from which temperature was estimated, assuming CBE sensitivity to temperature was $0.3 dB/^oC$. From Basu 2010 [10].

2.3 Summary

Ultrasound is an attractive modality for non-invasive, volumetric temperature estimation. Being cheap, portable and non-ironizing, ultrasound proved to be a promising replacement for MRI, the current *de facto* standard of temperature imaging [16]. Several temperature dependent parameters of ultrasound have been studied over the recent years, and we found that CBE is the parameter that is highly dependent on temperature in hyperthermia temperature range, and requires little prior knowledge.

The previous simulation studies and experimental work all have proved the promising feasibility of using CBE for non-invasive thermometry during thermal therapy. For clinical application of CBE temperature imaging, however, we need to be able to produce the temperature map of the ongoing therapy in real time, to effectively guide the thermal treatment. Our goal for this study is to implementing such a system, and reduce the motion compensation time to less than 1 sec.

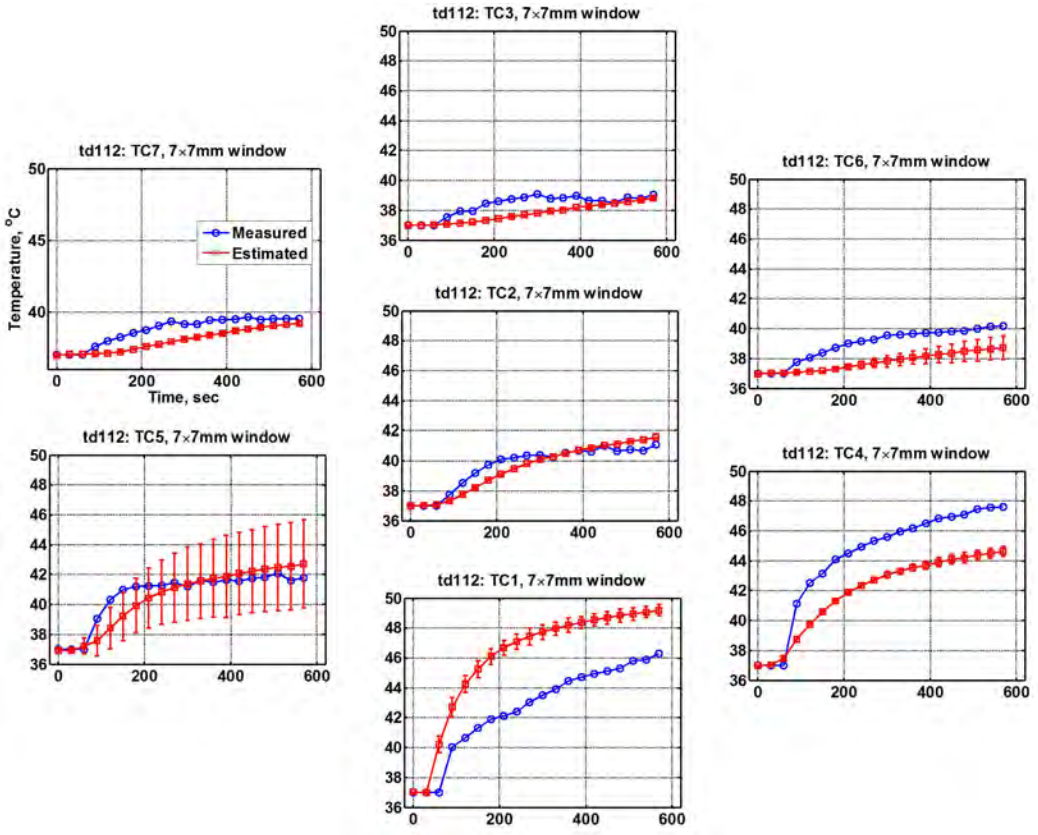


Figure 2.6: Estimated temperature versus temperature measured using thermocouples in turkey breast muscle (experiment td112). Measured temperatures were computed by averaging the two thermocouple measurements on each side of the 3D ultrasonic volume at the 7 locations. From Basu 2010 [10].

Chapter 3

Image Acquisition for Temperature Imaging

From the previous chapter, we learned that ultrasonic temperature imaging using Change in Backscattered Energy (CBE) is a promising approach to monitor temperature spread during thermal therapy. With its advantages in portability, noninvasiveness, accuracy and inexpensiveness [32], ultrasonic temperature imaging would not only provide good real-time information about the thermal distribution, which can help prevent damage to healthy tissue, but it can also sustain a thorough documentation of the thermal dosage and its effect in a treatment session, hence contribute valuable reference for future improvement [10].

It's known that CBE changes monotonically in accordance to temperature in hyperthermia temperature range (37^oC to 50^oC) [32]. To obtain precise temperature reading in clinical environment, a crucial factor is the temperature sensitivity of CBE. Our group has already gained abundant knowledge through years of studying on CBE in tissue regions with multiple scatterers, isolated individual scatterers, and in collections of individual scatterers.

In this project, we reproduced Basu's 3D CBE-Temperature dependence study in 2D. Furthermore, we implemented a two-computer architecture to realized real-time temperature imaging. In this chapter, we discussed the experimental setup of our ultrasonic temperature imaging system during non-uniform heating.

3.1 Calibration of CBE Thermal Sensitivity: Uniform Heating in a Water Tank

This section is a review of the previous calibration work done by our group, which included the thermocouples accuracy calibration, and CBE thermal sensitivity calibration.

3.1.1 Thermocouples Calibration

The internal thermistor of ThermoHaake (HAAKE Phoenix II, Thermo Fisher Scientific Inc., Waltham, MA) was used for the calibration of the thermocouples (OMEGA Industrial Hypodermic Probe, Hyp-3, OMEGA Engineering INC., Stamford, Connecticut). The thermistor temperature readings were first matched with respect to a NIST traceable digital thermometer (Fisher Scientific) to within $\pm 0.1^oC$ for all temperatures. During the calibration, all thermocouples were set below the thermistor under deionized and degassed water, heated by ThermoHaake circulating heater [10]. Readings of the thermocouples and internal thermistor were taken simultaneously every 0.5^oC interval from 35^oC to 52^oC.

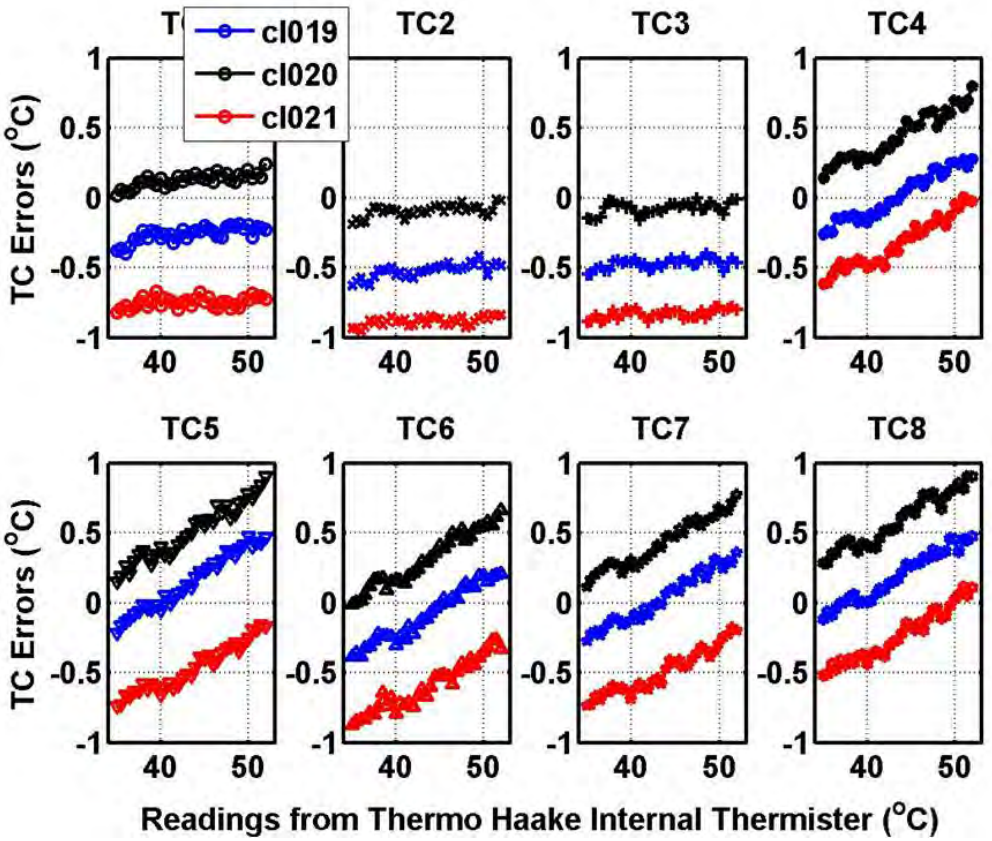


Figure 3.1: Thermocouple slope and offset calibration. Errors in the thermocouple readings with respect to ThermoHaake internal thermistor. The slope of the errors for each thermocouple was used in the subsequent experiments to correct the readings of each thermocouple. From Basu. 2010 [10]

Error data between thermocouple and ThermoHaake internal thermistor were obtained over 3 different experiments. Fig. 3.1 shows the error plot of each thermocouple, through which we can tell that each thermocouple reading error were increasing monotonically as opposed to temperature, and the slope was constant through out all 3 experiments. Whereas the offset of the reading error changed per experiment.

With that known, we can now get accurate temperature of thermocouple tip by applying the error slope, the thermistor temperature and error offset (retrieved by running thermocouple

calibration in the beginning of each experiment) to actual thermocouple readings and get accurate tip temperature.

3.1.2 Calibration of CBE-Temperature Dependence Parameters

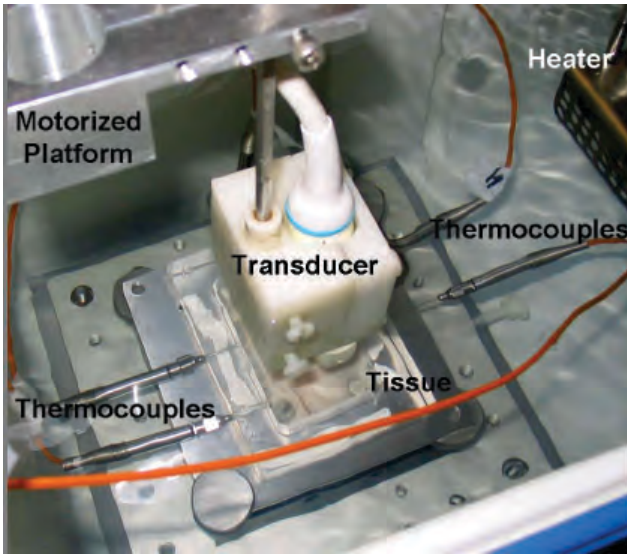


Figure 3.2: Experimental setup for ultrasonic temperature imaging. Water was heated uniformly to set tissue temperature. Four thermocouples determined when thermal equilibrium occurred in the specimen at which time ultrasonic images were taken. From Arthur et al. 2010 [3]

Fig. 3.2 illustrates the experimental setup for the CBE calibration work. Tissue sample was placed in the focal zone of a focused, phased-array transducer with center frequency of 7.5MHz. Both the tissue and the transducer were submerged under deionized, degassed water. Four calibrated thermocouples were inserted into four sites of tissue to record temperature for establishing equilibrium and calibrating CBE with temperature [5].

With water being heated by ThermoHaake from 37^oC - 45^oC, ultrasonic images were taken by the transducer every 0.5^oC during 9 uniform heating experiments. The tissue motion in each data set were found between adjacent frames and accumulated to get the motion of

each frame relative to the reference image, then compensated non-rigidly. CBE curves of each experiments were generated based on the motion compensated images.

A first-degree polynomial was created to fit the CBE values and the corresponding temperature readings. Since CBE at the reference temperature (37^oC) should be zero, an offset was added to this first-degree polynomial to make sure all CBE curves were zero at 37^oC. The slope of this first-degree polynomial represents the CBE-temperature sensibility [10].

3.2 Temperature Imaging: Estimation of Temperature during Non-Uniform Heating

Non-uniform heating was studied with two heat sources: 1) A soldering iron whose tip reached temperatures above 120^oC and 2) A hot-water tube with water at 75^oC.

3.2.1 Soldering Iron

To prove ultrasound temperature imaging can be a practical method in clinical circumstances, we designed an experimental setup that resembles the real life thermal therapy as shown in Fig. 3.3 (Left). With soldering iron as heat source, sticking into the tissue from the side of the donut plates.

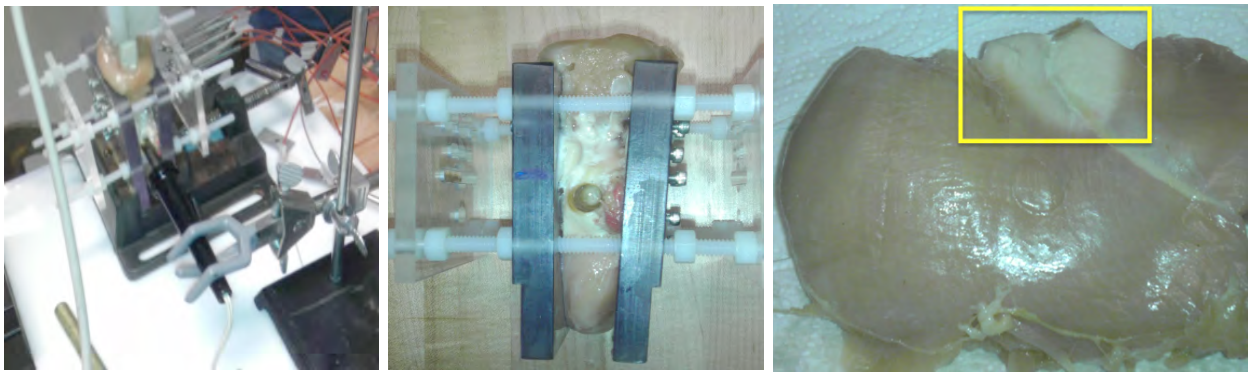


Figure 3.3: Left) Soldering iron inserted into the tissue, which was mounted in the donut fixture. Center) A view of the tissue where the soldering iron was in contact. Right) Front view of the tissue after the experiment, showing a region with denatured protein due to the heating.

Considering the fact that tip temperature of the soldering iron reached above 120^oC, to make sure it wouldn't melt the donut fixture, the soldering iron was inserted in tissue the way shown in Fig. 3.3 (Left) holding by a clamp. The tissue was fully cooked during experiment as seen in Fig. 3.3 (Right), yellow rectangle highlighted area. The denaturation created space between tissue and soldering iron and the way soldering iron was held caused the soldering iron tilting down gradually with the experiment went on.

Using soldering iron as heat source was aborted for this project due to the mechanical instability. Instead we turned to the hot-water tube setup, which is a mechanically stable structure, and the temperature of water can be set to any desirable value which gives us more control over the experiment temperature.

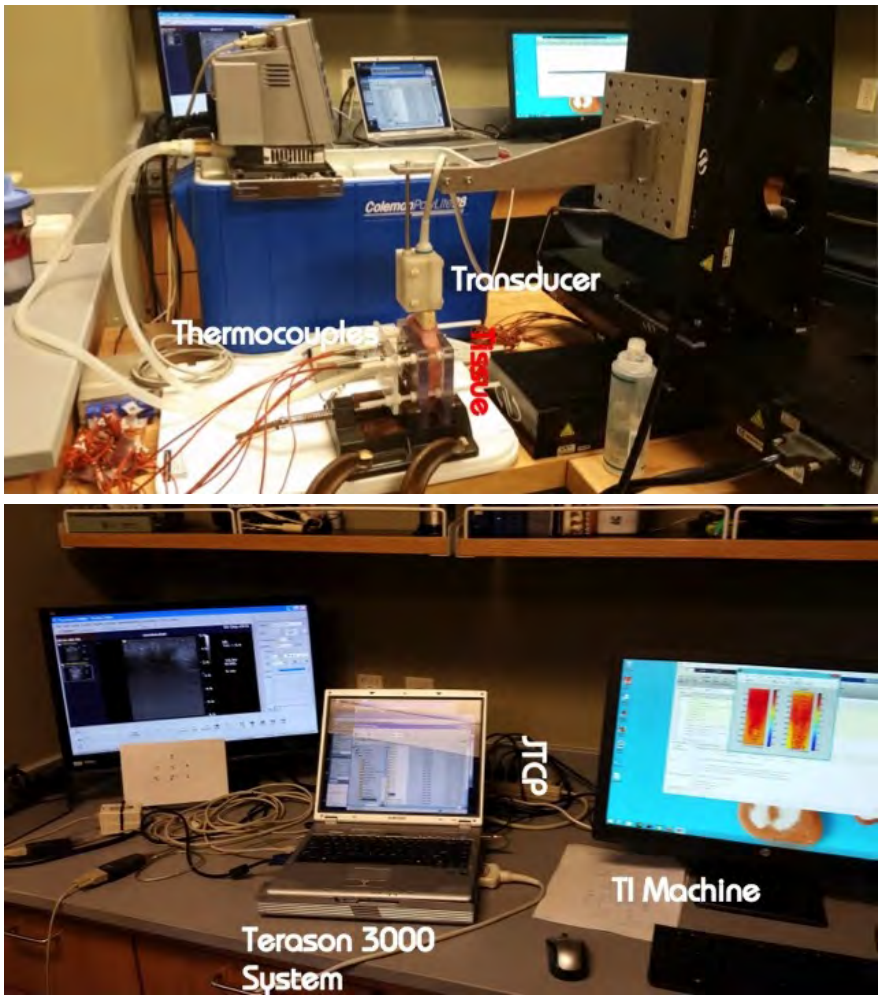


Figure 3.4: View of the experiment setup for non-uniform heating with a hot-water tube as heat source that shows the ultrasonic imaging system with its internet connection to the temperature imaging (TI) system. From Arthur et al. 2014 [4]

3.2.2 Hot-water Tube

The experiment was done on abattoir turkey breast muscle specimens heated by a silicon hot-water tube as shown in Fig. 3.4. The full code use on the Terason 3000 is listed in Appendix A. A donut fixture designed specifically for this experiment was used to hold the specimen, it has customized holes on both of the donut plates, which allow the water tube to go through. To minimize any possible motion during experiment, the donut fixture was

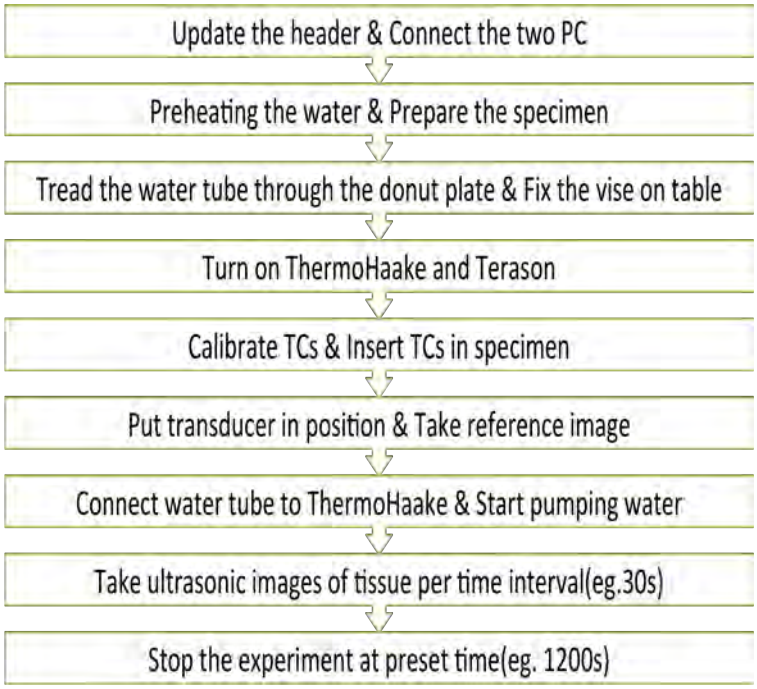
clamped on table. The temperature at 7 different locations on specimen was measured with 7 thermocouples inserted in the tissue through the screws on the supporting bracket in Fig. 3.7 (Lower right). Temperature readings of thermocouples were obtained by the Dataq (DI-1000-TC-8, DATAQ Instruments Inc., Akron, OH) data acquisition box. The thermocouples had an inherent error of $\pm 0.1^{o}C$ [10].

Temperature of water run through the water tube was set by controlling a circulating heater (HAAKE Phoenix II, Thermo Fisher Scientific Inc., Waltham, MA). With hot-water pumping through the silicon water tube, we get a steady heat source in the center of the tissue [3].

The specimen was placed in the focal zone of a 128 element (each element has a frequency of 7.5MHz) linear transducer array (model 12L5) focused at 4.5 cm, center of the tissue specimen. NewPort IMS300PP stages controlled by ESP300 controller (Newport Co., Irvine CA) was used to move the transducer to the desired location. Terason 3000 system (Teratech Corp., Burlington, MA) acquired the phase-array images taken by transducer every preset time step during the non-uniform heating.

The Terason computer which controlled the imaging system also controlled the stepper motor and the Thermal Haake via serial connections. Thus the experiment process was fully automated and controlled by a custom Matlab program that run on the Terason notebook, all we need to do is the preparation before the experiment starts and some key-strokes to switch between Matlab and the Terason 3000 applications to save image files. All key-strokes needed were realized by autoIT keystroke-emulation software (hiddensoft.com) [3] [10].

In order to make more reliable imaging, we implemented control of Terason Imaging system using the software development kid (SDK) from TeraTech. The SDK allows us to access Terason image stream via AcitveX control in our Matlab control program. Before create ActiveX control, 3 .ocx files should be registered: *Regsvr32 TTFrameReceiver.ocx; Regsvr32 TTAutomate.ocx; Regsvr32 TTSimpleImageWnd.ocx.* 3 functions for acquiring images using the SDK were programmed by our team: 1) *hTTauto = StartTerasonActx(exam).* 2) *savesglimage_ttauto(hTTauto,filename).* 3) *saveloop_ttauto(hTTauto,filename).* The scripts of these function can be found in Appendix A. The above 3 functions are simple applications only for our current experiments [15].




```
THERMOMETRY HEADER          V3
EXPERIMENT ID (5 DIG)   TD709
DATE DDMMMYYYY      10JUN2014
IMAGING SYSTEM Terason     3K
EXAM FILE          cbedonutair
TRANSDUCER Accuson        12L5
NUMBER OF ELEMENTS         128
FOCUS (mm)                  30
SAMPLE RATE (MHz)           24
CF (MHz)                   7.5
SOS (mm/us)               1.54
PERSISTENCE                  3
EXPERIMENT
HOT WATER TEMPERATURE     75.0
IMAGE TIMES          0:30:1200
TCCAL               tcs116.mat
TCOUPLE IN WATER           1:2
TCOUPLE IN TISSUE          3:8

Tissue:          turkey breast
bought      sell by 08JUN2014
Medium:                    gel
Program:     bldonut_zwy8sv.m
Notes:  Justify @ column  30
```

Figure 3.5: Left) Flowchart of the control the experimental process. Right) Header in which all parameters used in the experiment were defined.

Experimental Procedure

```
7/18/14 12:51 PM          MATLAB Command Window                    1 of 1

**EXPERIMENTAL SETUP**
Clamp the vise to the table
Adjust the distance between the thermocouple supporting
  bracket and donut plates to 2.5cm inner surface to inner surface
Drill a 5 mm hole (cork borer) in the tissue center
Put the tissue between the donut plates
Hand tighten the nylon screws
Use long screw in hot water tube to thread tube through tissue
->Use gel for tissue contact
Put donut assembly in the vise and tighten vise

7/18/14 12:52 PM          MATLAB Command Window                    1 of 1

**IMAGING**
Fix the transducer to the motor mount
Put coupling gel on the tissue
Move the transducer to the tissue with the NewPort
IF CONNECTED TO EXPERIMENTAL DATA DIRECTORY, PRESS ANY KEY
```

Figure 3.6: Instructions from the custom Matlab code for controlling the experiment preparation and execution. The Matlab code for this part can be found in Appendix A.

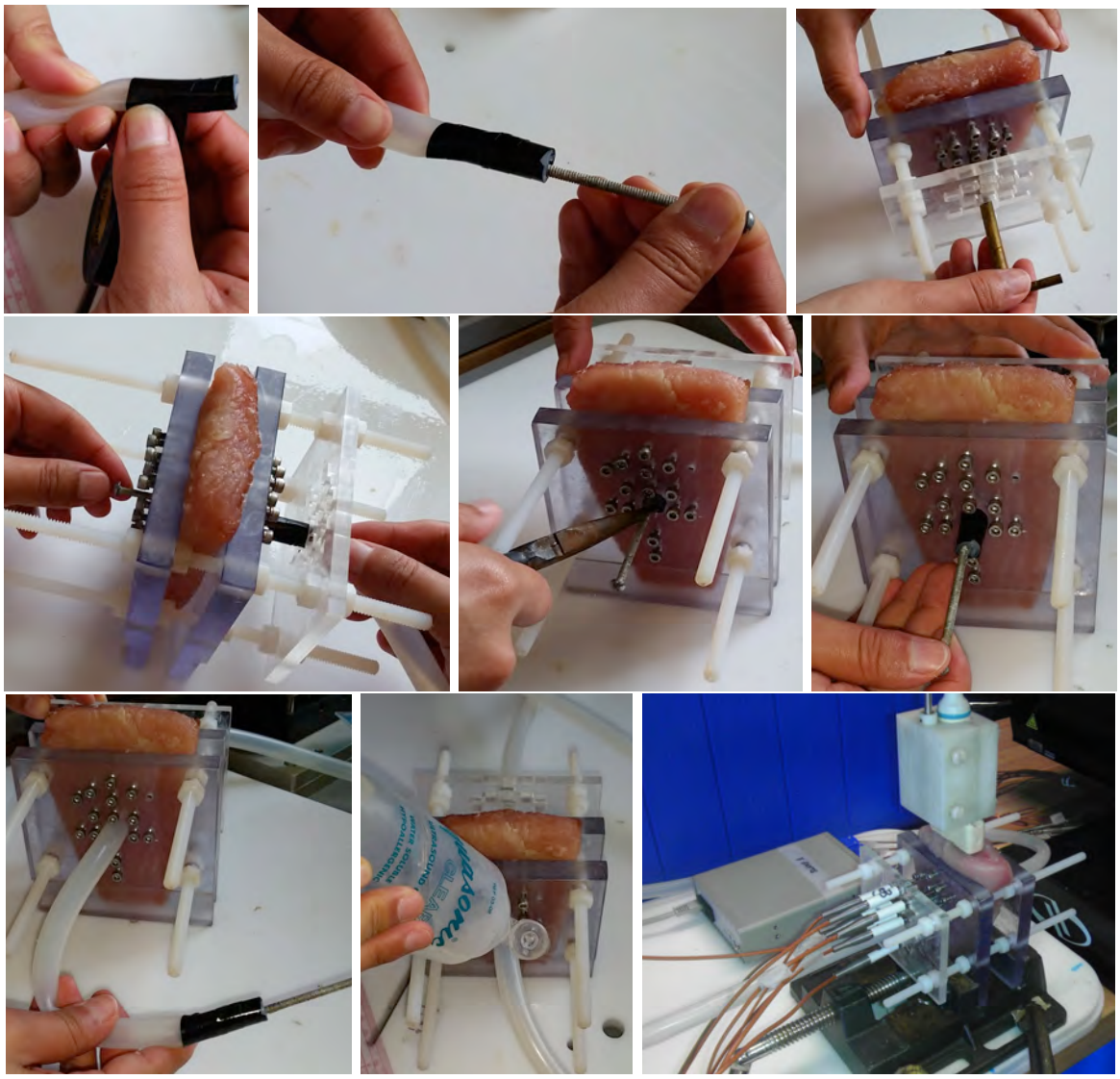


Figure 3.7: Steps for inserting the hot-water tube into the tissue. Description of the steps shown above is given in the text. Lower Right) Fixture, tissue with hot-water tube, ultrasonic transducer and thermocouples just before heating was begun. Thermocouples were inserted into the tissue through a support bracket to reduce motion.

The flow chart as shown in Fig. 3.5 contains every step of a non-uniform heating experiment. In the beginning of every experiment, we create a header file specifically for the experiment. The header file 3.5 (Right) of the each experiment includes all the preset parameters and informations about the experiment, such as time range, time interval, water temperature,

experiment ID and the tissue properties. With these variables predefined, the custom controlling Matlab program can read them directly from the header in the beginning of every experiment, and we can use one set of custom Matlab routine for all the experiment, without spending time on code modification every time.

An important part of the experiment is the preparation work for the experimental setup, the lab personnel should get every step done right, one mistake could lead to both a huge waste of time and unusable data. Therefore we implemented the instructions for every preparation step in the custom Matlab routine as shown in Fig. 3.6, which guides the lab personnel step by step till the start of image acquisition.

To obtain a more obvious temperature spread in the tissue, we decided to set the water temperature to 75^oC. It will take about an hour for Thermal Haake to heat the room temperature water to 75^oC. To improve the experiment efficiency, the water in tank was preheated for 30 minutes before the preparation work began. As the Thermal Haake can get the heating done with no supervision, this portion of time can be spent more cost-effectively.

The vital part of the preparation work is to get the water tube (with an approximate 1.5cm outer diameter) through a 0.5cm tissue hole. Fig. 3.7 documented the detailed instructions to get this step done. First put tissue in between the donut plates, and hand tighten the nylon screw to keep tissue from moving. Then use a 0.5 cm diameter cord to cut a hole in the tissue through the round holes on the donut plates. Tighten the tip of the water tube with tape, then push a long screw into the tip of tube, make sure the tip is very tight so that the friction between the screw and the silicon can be maximized. Thread the long screw through the hole in tissue and pull the screw to get the water tube through the tissue as well, in the mean time, a little pushing from the other side of the donut plate would help a lot. When the tape wrapped water tube tip can be seen, the plier could be used to get the rest of the pulling done.

Last but not least, smear coupling gel evenly on the water tube, and stretch the tube so that the part with gel will be right inside the tissue. This gel made sure there was no air between the tissue and the silicon tube and we can get a clear view of the tube in tissue in ultrasonic images. Finally, couplers were put on each end of the water tube and made ready to connect to the Thermal Haake. Fig. 3.7 (Lower Right) shows how the donut fixture looks like after all the previous preparation work. We can see the setting of donut fixture, tissue

with hot-water tube, ultrasonic transducer and thermocouples before the heating started. Thermocouples were inserted into the tissue through a support bracket to reduce motion.

To achieve real-time TI, a two-computer architecture was applied. Our Terason 3000 ultrasonic imaging system collected and sent raw images over a jtcp internet connection to a computer, which estimated temperatures images. The TI machine was an HP Envy Phoenix 810 with a GeForce GTX 770 GPU card. The TI computer (RTTI) performed motion compensation and extracted temperature images. As displayed on Fig. 3.8 (Right) and Fig. 3.9, the RTTI system monitor displays the correlation coefficient between each motion compensated ultrasonic image and the reference image, along with the processing time.

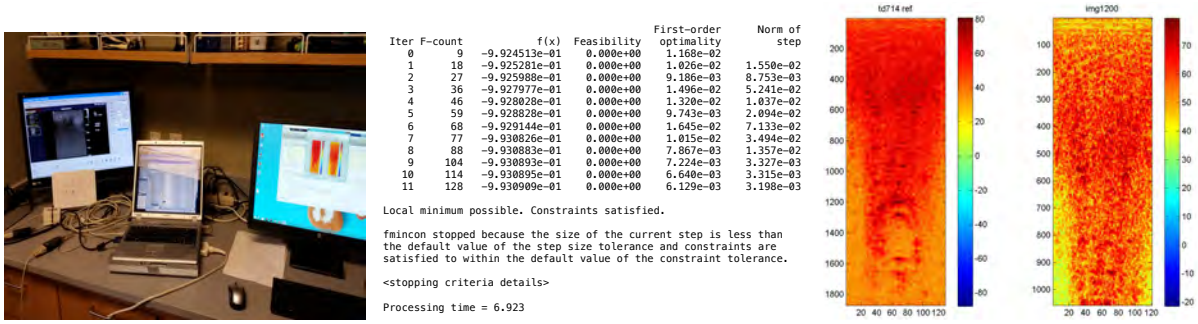


Figure 3.8: Left) Terason3000 system image of the the specimen in the custom holder. Center) Motion compensation processing during real-time analysis of each ultrasonic image. Bottom line shows the total analysis time for the given image (6.923 sec). Right) Reference image compared to the current motion compensated image.

We used non-rigid motion compensation method on RTTI machine to get the image shown in Fig. 3.8 (Right). In the beginning of the project, we were applying rigid method Fig. 3.9, because we thought that would be more time efficient. With a tryout with non-rigid method, however, proved it can rendered a more ideal motion compensation result with limited compromise in time, which will be discussed in detail in Chapter 5. Thus we modified our custom Matlab code to do real-time non-rigid motion compensation.

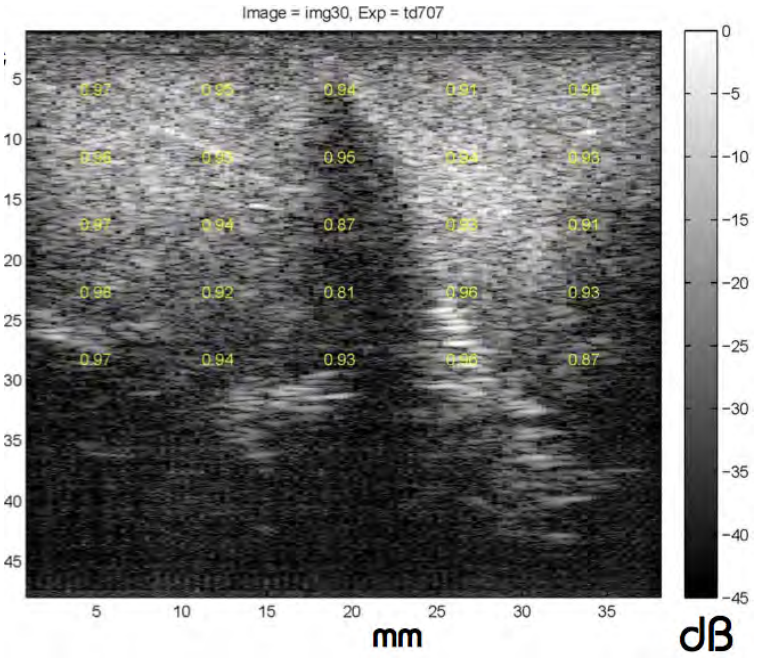


Figure 3.9: View of the temperature imaging (TI) computer screen which received and processed ultrasonic images in real time. The numbers on the image are correlation coefficients after rigid motion compensation.

In this study, we used thermocouple readings as the standard to calibrate our CBE estimated temperature. Therefore, it's important to identify the exact coordinates of the thermocouples on a 2D ultrasonic image. Thermocouple locations were decided through a bitmap image as shown on Fig. 3.10 (Left), which was taken and saved by Terason 3000 system while the thermocouples were still in image. The yellow-rectangle highlighted spots are the positions of thermocouples. Fig. 3.10 (Center) shows that the sites were also marked on monitor with a colored water marker, in this case, pink.

With the thermocouple reading and the CBE calculated at the corresponding thermocouple spot, we can calibrate our CBE-Temperature sensitivity. Temperature of thermocouples varies according to their locations, those closer to the hot-water tube had a larger increase in temperature, whereas those located further to the tube showed very little temperature change as seen in Fig. 3.10 (Right).

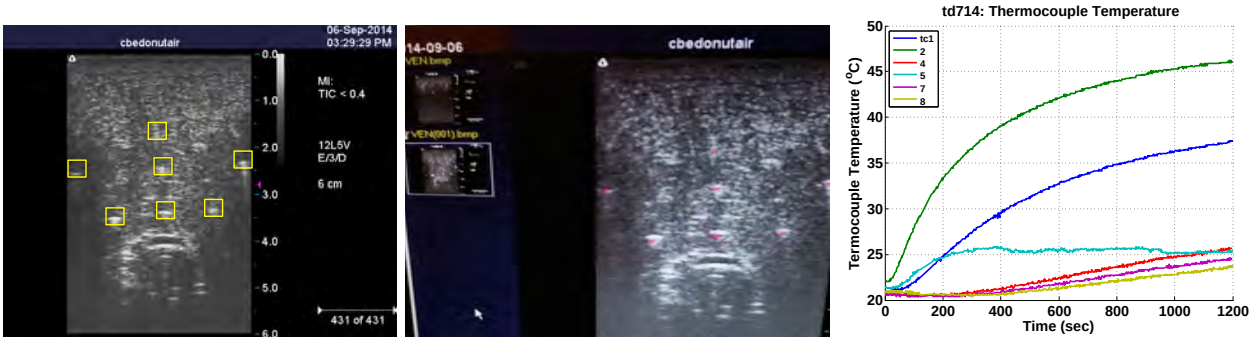


Figure 3.10: Left) Terason image with thermocouple locations highlighted with yellow rectangles. Center) Red marker was used to pinpoint the thermocouple locations. Right) Thermocouple readings as a function of experiment time.

3.3 Summary

In this chapter, we reviewed the previous calibration work done on CBE thermal sensitivity by our group, summarized how the thermocouples were calibrated to serve as a foundation for the CBE calibration work, and the process to get the CBE-Temperature parameters (slope and offset of the CBE curve) which we are going to apply directly in data analysis of Chapter 4. The second part of the chapter, contains our experimental procedure for non-uniform heating with hot-water tube, with details of the how the two computers work in synchronization to control the experimental procedure as well as analyzing data in real time.

We also discussed how soldering iron as an alternative heat source was not a stable setup, and documented core instructions for the preparation work. In the future, we plan to run the custom Matlab code on the RTTI machine using its GPU, with microwave antenna as heat source to simulate clinical thermal therapy with real-time ultrasonic temperature imaging system.

Chapter 4

Motion Compensation & CBE Temperature Images

Chapter 3 provided a detailed documentation of the experimental setup for 2D ultrasonic temperature imaging system with hot-water tube as the heat source. Compare to the previous 3D study done by Basu [10], the 2D system requires a simpler experimental setting and less data acquisition. The 2D system also eliminated the motion introduced by moving the transducer in the elevation direction, hence reduced the time consumed by motion compensation (MC). Therefore, we decided to carry out the real-time temperature imaging study in the 2D ultrasound system.

In this chapter, we proved that the 2D setup still maintains a good CBE-Temperature accuracy compare to that of the 3D system. 6 experiments were carried out during this study, the turkey breast specimen at each experiment was heated in air from room temperature by a hot-water tube for 1200 sec. 75^oC water was running through the silicone hot-water tube the entire experiment. 2D ultrasonic images were taken every 30 sec by the transducer. The experimental data along with B-mode images obtained during experiments were analyzed using custom Matlab routines. The analysis including motion compensation, correlation coefficient calculation, CBE calculation and the estimation of temperature using CBE.

The 6 experiments are: td709, td710, td711, td712, td713, td714, among which there are good experiments and also less than ideal ones. They are analyzed individually in this chapter to demonstrate the CBE-Temperature estimation sensitivity in different scenarios.

4.1 Reference Images

Below are the B-mode reference images of each experiment with custom colormaps on. Fig. 4.1 provided an intuitive illustration about the quality of each experiment. We can see from the bright spots right above the tube that the reference image of td709 was taken with thermocouples left in the view. We can also tell that td713 didn't have a good ultrasonic penetration.

The actual experiment setup does support those intuitive conclusions. For td709, we forgot to pull the thermocouples out the view before we started the experiment, and they were left there during the entire experimental time. As of td713, we left the experimental turkey abattoir in room temperature for 12 hours to achieve temperature equilibrium in tissue, however, without enough precautions, the tissue was severely dehydrated, and thus caused the poor penetration of ultrasound signals in the turkey specimen.

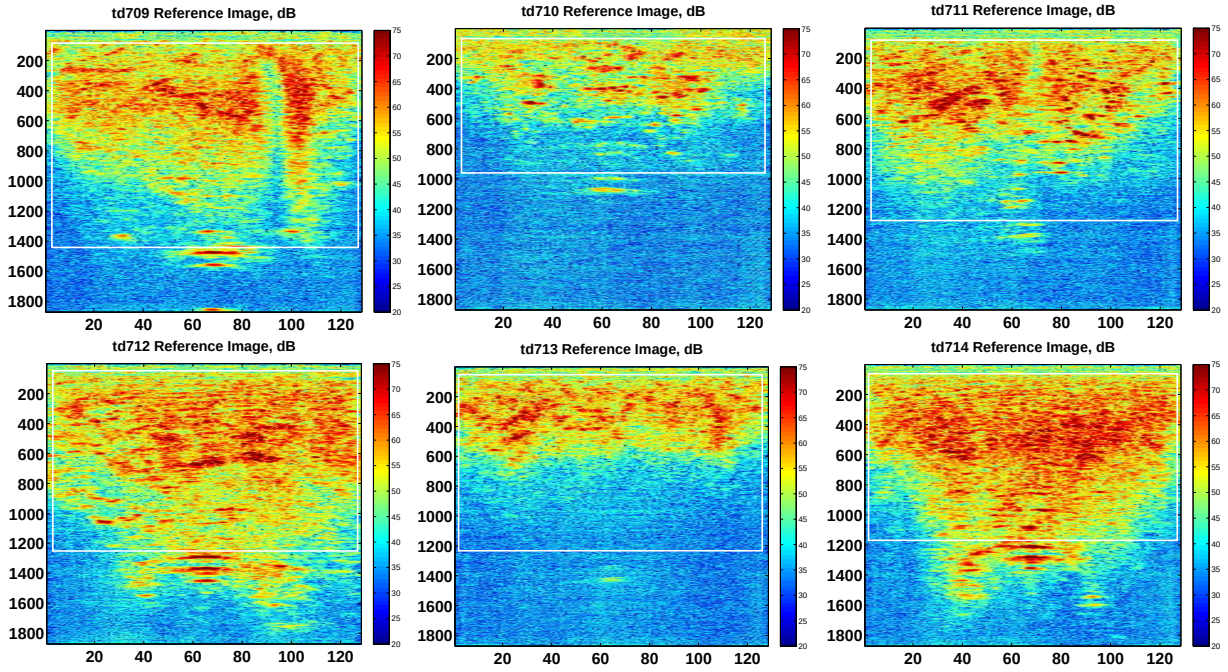


Figure 4.1: Reference images in B-mode for experiments td709, td710, td711, td712, td713 and td714. The white rectangle in each image represents the Region of Interest(ROI) for CBE temperature imaging.

4.2 Motion Compensation

Motion compensation is a crucial part of the temperature estimation using CBE. Because in our initial work, we found that un-compensated motion in images can lead to spurious CBE, which will affect the accuracy of the estimation [15]. In this study, we are using an algorithm developed by Dr. Trobaugh for non-rigid motion estimation and compensation over region of interest (ROI). The motion field was modeled to change linearly over ROI and represented as a linear function of the motion at the control points, which were chosen as the 4 corner points of the ROI. Motion was obtained through searching the displacements at the control points that maximizes the cross correlation between two images. Then interpolate the shift at 4 corners to the entire ROI to get the motion field. For a sequence of images obtained in experiments, motion was estimated between adjacent image pairs and then accumulated to the reference image [15]. 2D motion field is demonstrated by Fig. 4.2, with arrows at 4 corners of region show the value and direction of the shift.

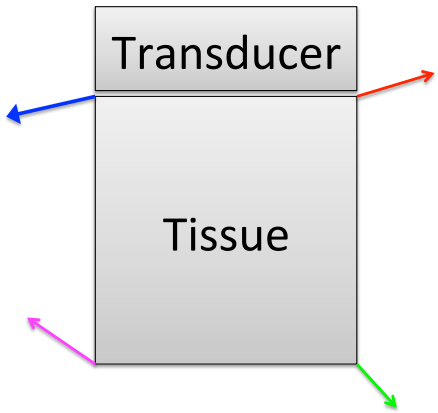
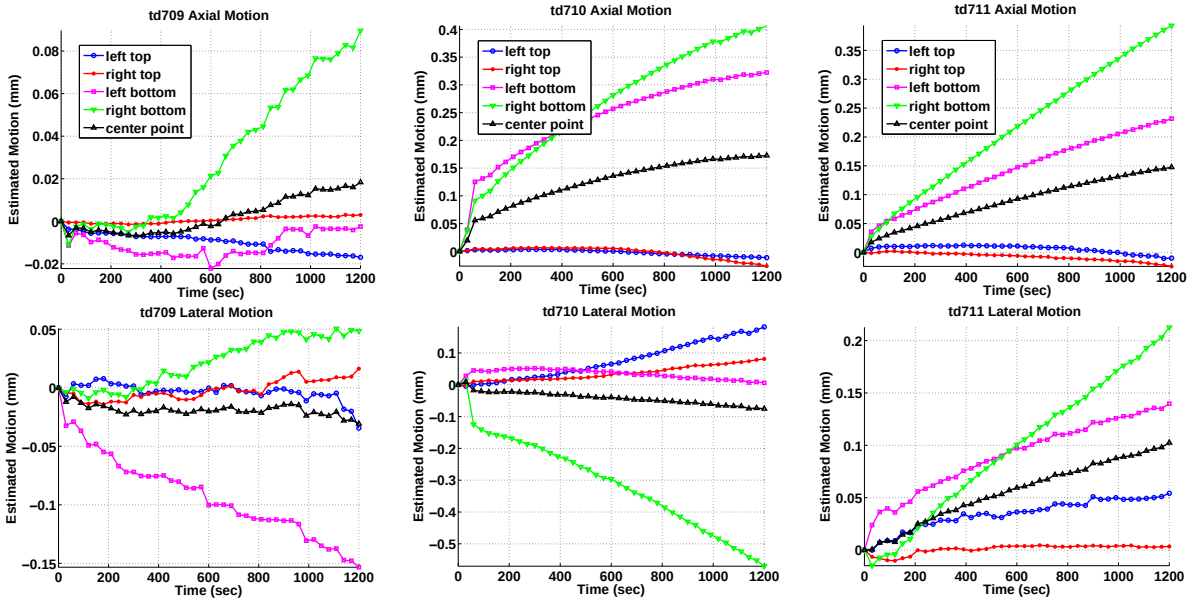


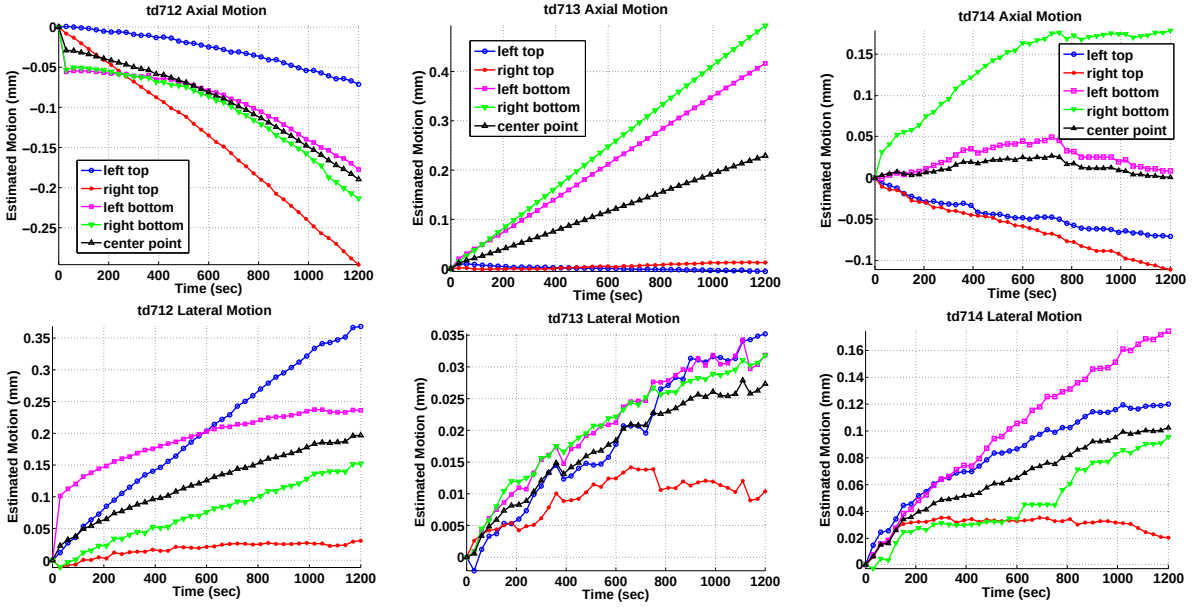
Figure 4.2: Motion compensation using estimated motion at four corners of a 2D image. The arrows represent direction of motion and amount of motion. Colors used in each corner correspond to colors used in Fig. 4.3

4.2.1 Motion Accumulation in Axial and Lateral Direction

Fig. 4.3 shows both axial and lateral motions at the image corners of the ROI (Fig. 4.2). As shown in Fig. 4.3, accumulated motion in all experiments was < 0.5 mm.



(a) td709 (b) td710 (c) td711



(d) td712 (e) td713 (f) td714

Figure 4.3: Axial and lateral motion at four corners and the center point of each 2D image region (ROI). The motion value is plotted in mm as a function of experiment time. As shown in this figure, the largest motion in all experiments is less than 0.5 mm.

4.2.2 Quiver Motion at the Final Frame

Motion at the last frame of each experiment was illustrated in quiver at 25 selected sites in ROI, as seen in Fig. 4.4. The arrows represent both direction and value of the motion accumulated through the experiment time. Values of the motion were multiplied by 50, to ensure the visibility in the scale of the ROI.

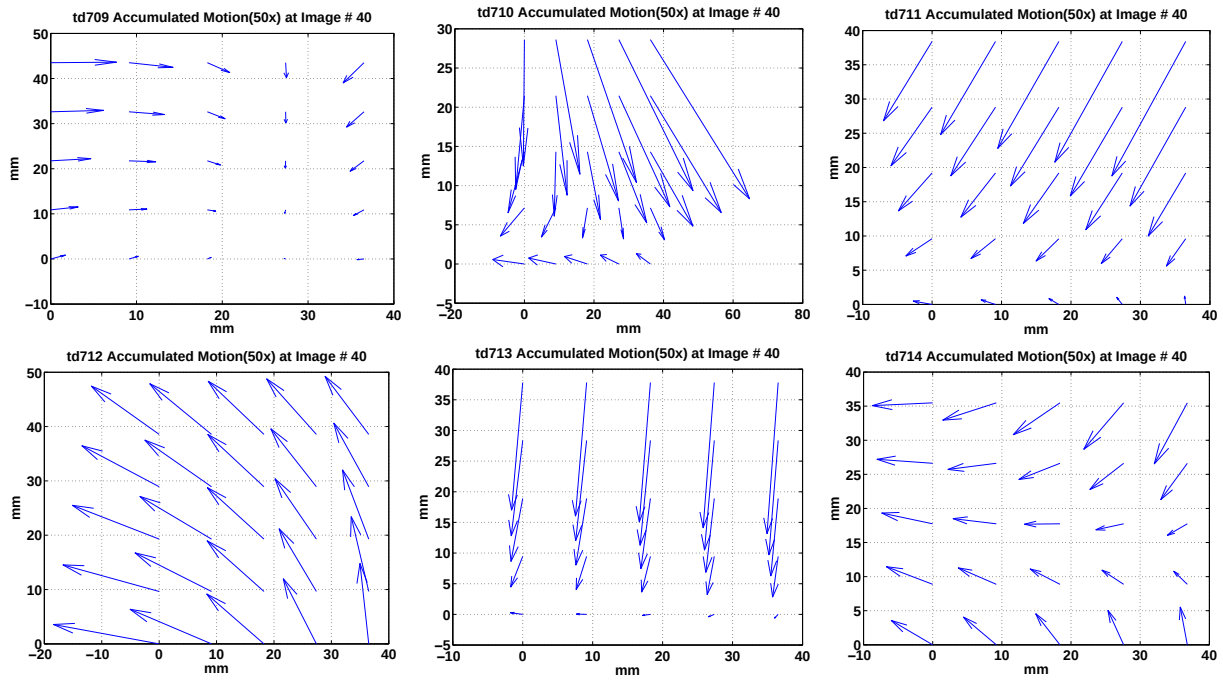


Figure 4.4: Accumulated motion by the end of each experiment (after 20 minutes) at 25 distinct sites in ROI. Quivers represent both the value and direction of motion on those sites. Value of motion is enlarged 50 times for visibility on the same scale of the ROI.

4.2.3 Correlation Coefficient Before and After Motion Compensation

Fig. 4.5 shows the correlation coefficient before and after motion compensation (MC), between adjacent image pairs and between reference images and other images. As we can see from the figure below, correlation coefficients (CC) for each experiment was above 0.99 between adjacent image pairs both before and after MC.

Whereas CC between reference images and other images can be as low as 0.75 (td712) before MC, and 0.77 after MC (td712), and that the CC values were decreasing with time. The decrease was caused by both motion and thermal change. We can see in Fig. 4.5 that the CC after MC is higher than that before MC, which means with the motion being compensated, CC decrease was purely affected by heat spread.

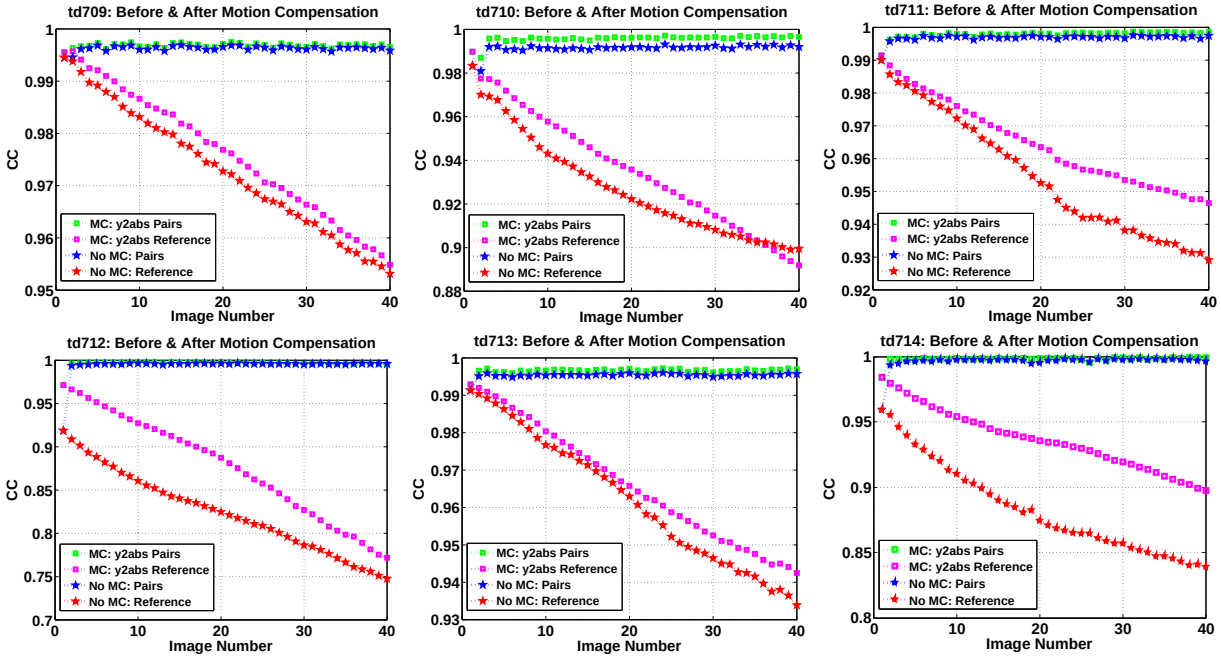


Figure 4.5: Correlation coefficients as a function of image number, before and after motion compensation between adjacent image pairs and reference image and other images.

The reason CC between reference images and other images is lower than between adjacent images is, with experiment time incrementing, temperature became a dominant factor affecting the correlation coefficient between two images. With motion no longer the major cause for in-correlation between images, MC would not have an as obvious effect as it did on adjacent image pairs.

4.3 Change in Backscattered Energy (CBE)

CBE values were calculated within the region of interest during the experimental time. We can see in Fig. 4.6 that the CBE value before motion compensation is higher than that after motion compensation for both positive and negative CBE, which in another way demonstrated how the motion would contribute to the CBE value, and the MC did help reduce spurious CBE.

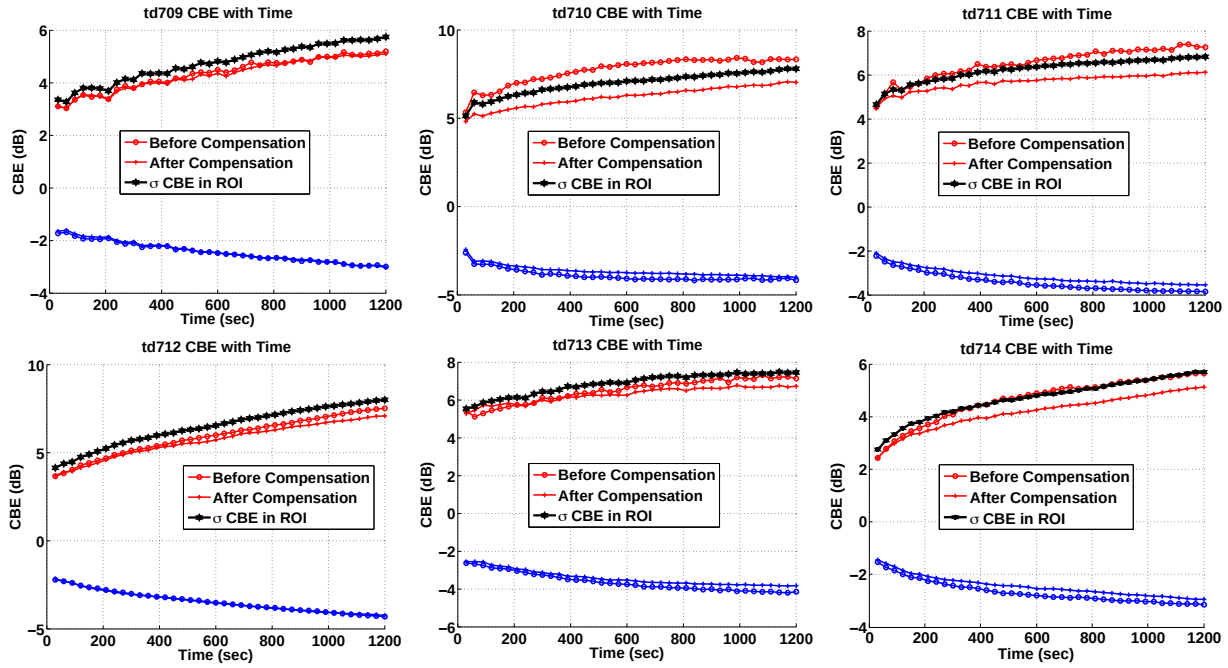


Figure 4.6: Positive CBE (PCBE), negative CBE (NCBE) before and after motion compensation (MC), and the standard deviation of PCBE and NCBE after MC as a function of time over the ROI.

4.3.1 Thermocouple Temperature Increases Over Time

The thermocouple temperature readings at different tissue locations (Fig. 4.8) over experiment time are shown in Fig. 4.7. We can see that the closer the thermoucouple to the hot-water tube, the larger the temperature increase. We can also tell that thermocouple# 5 was not working properly except in td710. Because it's readings were very noisy, and didn't get more temperature increase than TC8, the most distant thermocouple from heat source.

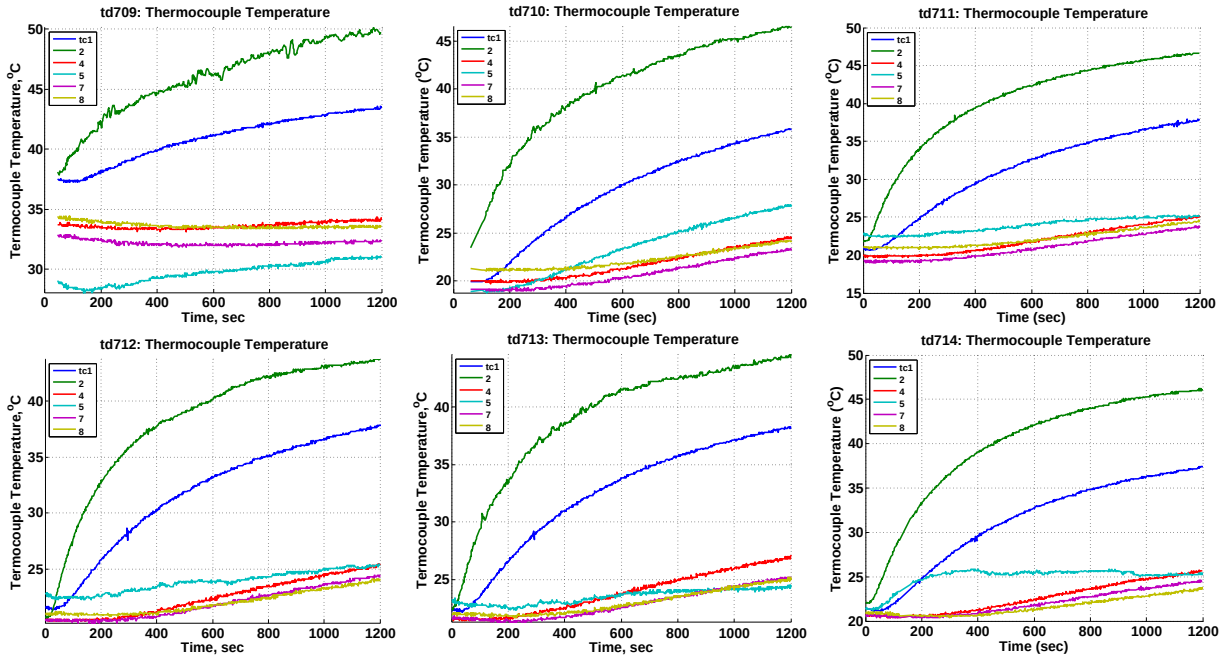


Figure 4.7: Thermocouple temperatures in different tissue locations as a function of experiment time. From the TC5 readings, TC5 was functioning properly only in td710.

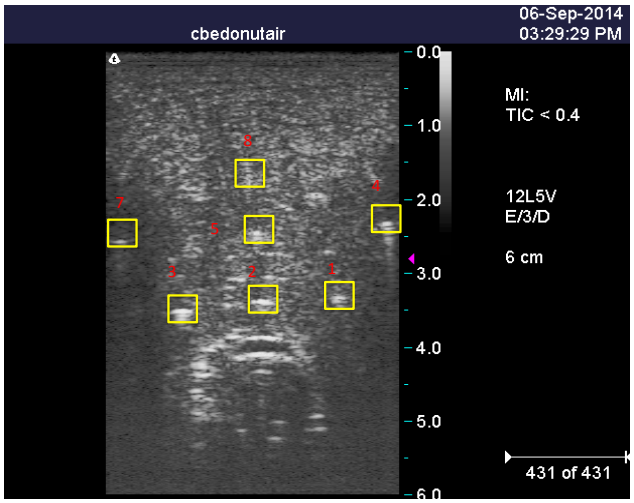


Figure 4.8: Thermocouple locations, labeled with thermocouple number, in a B-mode ultrasonic image taken before the experiment started. The relative position of thermocouples was the same in each experiment.

4.4 Successful Experiments: td709, td710 and td714

Compare to the other 3 experiments, td709, td710 and td714 were more successful because they were carried out more precautiously with no mistake in any experimental process, therefore yielded more trust-worthy data.

4.4.1 td709

Fig. 4.9 displays B-mode ultrasonic images of the reference image and the last image in td709. As shown in this figure, tc1, tc2 and tc3 were in the 2D imaging view through out the experiment. Below are the quiver motion figures of td709 at 0 sec, 600 sec, 900 sec, and 1200 sec. As listed in Fig. 4.10, we can see that the motion was accumulating with time. Thermocouples were used as the standard to evaluate the CBE sensitivity in temperature estimation. Before the heating started in each experiment, thermocouples were pulled right out of the 2D image view, to eliminate the possible error caused by ultrasonic interference

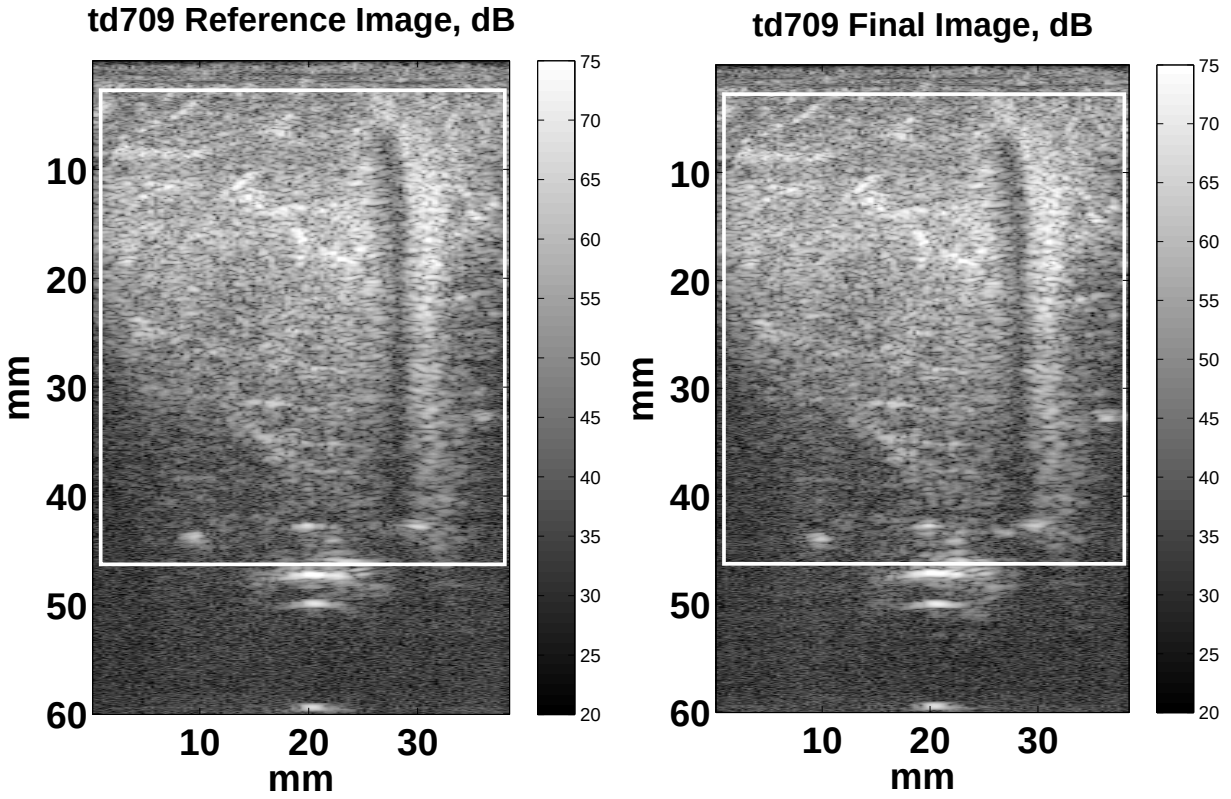


Figure 4.9: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image . Right) Last image of the experiment run (after 20 minutes).

of metal tips. Meanwhile they were close enough to the 2D image view, that their readings were eligible to represent the temperature at the corresponding sites in the view.

To calibrate CBE-Temperature sensitivity, finding the nearly accurate location of thermocouples after motion compensation is the key. We predicted the spots closer to hot-water tube would have a more accurate temperature estimation using CBE. Because they were exposed to the most thermal energy growth, and presumably would reduce the error introduced by noise and uncompensated motion.

Thus thermocouple (TC) #1, 2, 3 became the focus of our study. TC2 was not used because it was too close to the hot-water tube that the temperature gradient from the heat source could largely interference with the CBE calculation. Meanwhile it's hard to select a 6x6 mm^2 area centered at TC2 to calculate CBE at TC2 site without including part of the hot-water tube. With TC3 broken, TC1 turned to be the thermocouple we are most interested in.

We first located TC1 in the B-mode ultrasonic image taken before the heating started, with thermocouples in the view. The white rectangle with blue dot in center indicates the location of TC1 as shown on top left of Fig. 4.11.

Note that thermocouple location coordinates would change after motion compensation (MC), because the MC was done over ROI, which is smaller than the original image size. Therefore we need to take the x-offset and y-offset into account when we calculate the new thermocouple coordinates.

Meanwhile, to compensate the motion, the MC process itself could cause the coordinate shift as well. The shift is normally less than 20 pixels in y-direction (more than 1000 pixels) and 2 pixels in x-direction (about 120 pixels). Nevertheless, we don't have a method or equation to precisely calculate the shift caused by MC, so we used landmarks in images both before and after experiment started.

Through comparing the coordinates of the landmark before and after MC, and deducting the shift caused by lateral and axial offsets, we obtained the MC caused shift. Hence the nearly accurate coordinates of TC1 on motion compensated images were gained, as shown as red dot in the yellow rectangle on top right of Fig. 4.11.

Even though we made a mistake by leaving the thermocouples in the ultrasonic image acquisition view in td709, it actually helped us to prove the accuracy of our method of finding thermocouple location after motion compensation. Since we can easily tell the location of TC1 on the motion-compensated image, we can then compare it with the coordinates gained by our method.

In this case, the red dot on top right of Fig. 4.11 represents the TC1 position obtained from our calculation, and we can see that it lies exactly on the bright spot, which is TC1 on the motion-compensated image. With the accuracy proven, now our method can be safely applied to other experiments as well. With the TC1 location in motion-compensated images obtained, we can calculate the CBE values at that point and then get the CBE estimated temperature. The envelopes of motion-compensated image were found with Hilbert transform, then a 3x3 running average filter was applied. The 33 filter is a low-pass filter that removes outliers to reduce noise effects. To get the backscattered energy (BE), the corresponding values were squared at each pixel [3].

The CBE at TC1 site was calculated based on the 6x6 mm^2 [10] region centered at TC1. The CBE value in this 6x6 mm^2 region was selected from the previously calculated CBE over the entire ROI at each pixel. Thresholds were applied to filter out some noise while finding positive CBE (PCBE) and negative CBE (NCBE). The CBE at TC1 was the standard deviation between the mean of PCBE and NCBE.

The bottom center image in Fig. 4.11 shows the CBE value at TC1 as a function of experiment time. We can see that the entire CBE increase is about 2.5 dB. The initial jump of 3 dB was considered caused by noise.

CBE-Temperature sensitivity and offset were predicted using first-degree polynomial fit between CBE values and the TC1 temperature readings, with 35^oC as the initial temperature at TC1 according to Fig. 4.7. The TC1 temperature readings and CBE estimated temperature were shown on bottom left of Fig. 4.11. From the previous analysis, we derived that in td709, the CBE-Temperature sensitivity at TC1 is 0.206 $dB/^oC$, with an error of 0.5 ± 0.4^oC.

CBE value was also plotted against TC1 reading, as seen on bottom right of Fig. 4.11. The red line (with slope = 0.3 $db/^oC$ [10]) was used as a reference for the CBE-Temperature relationship.

CBE images were created at frame #1, 10, 20, 40 on the same color scale. Fig. 4.12 illustrates how the CB energy spreads with time.

When the 0.3 $dB/^oC$ CBE-Temperature sensitivity was applied to the CBE values (with the 3.24 dB offset deducted), along with the initial temperature of 35.3^oC, we got the temperature images at frame #1, 10, 20, 40 in Fig. 4.13.

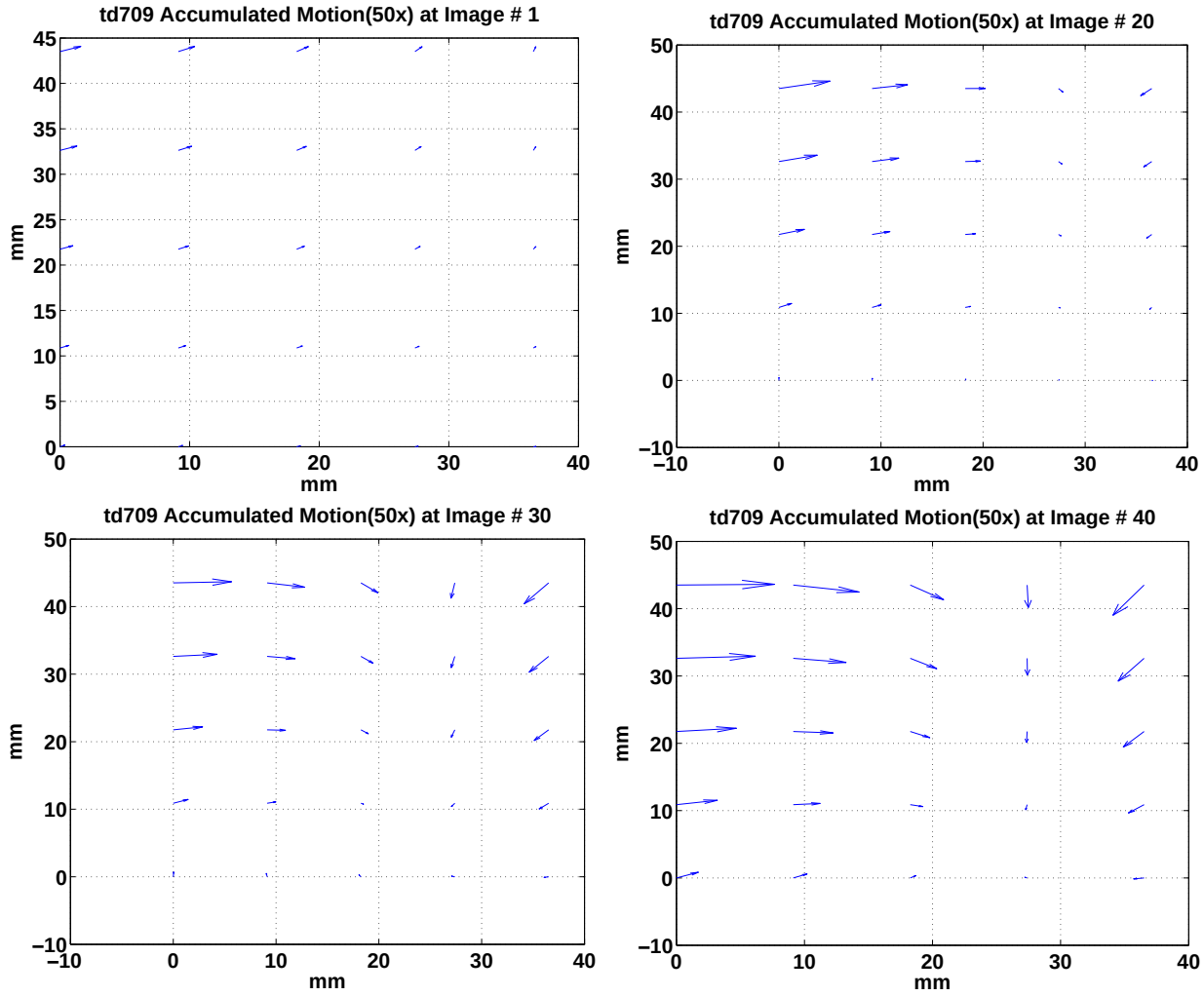


Figure 4.10: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

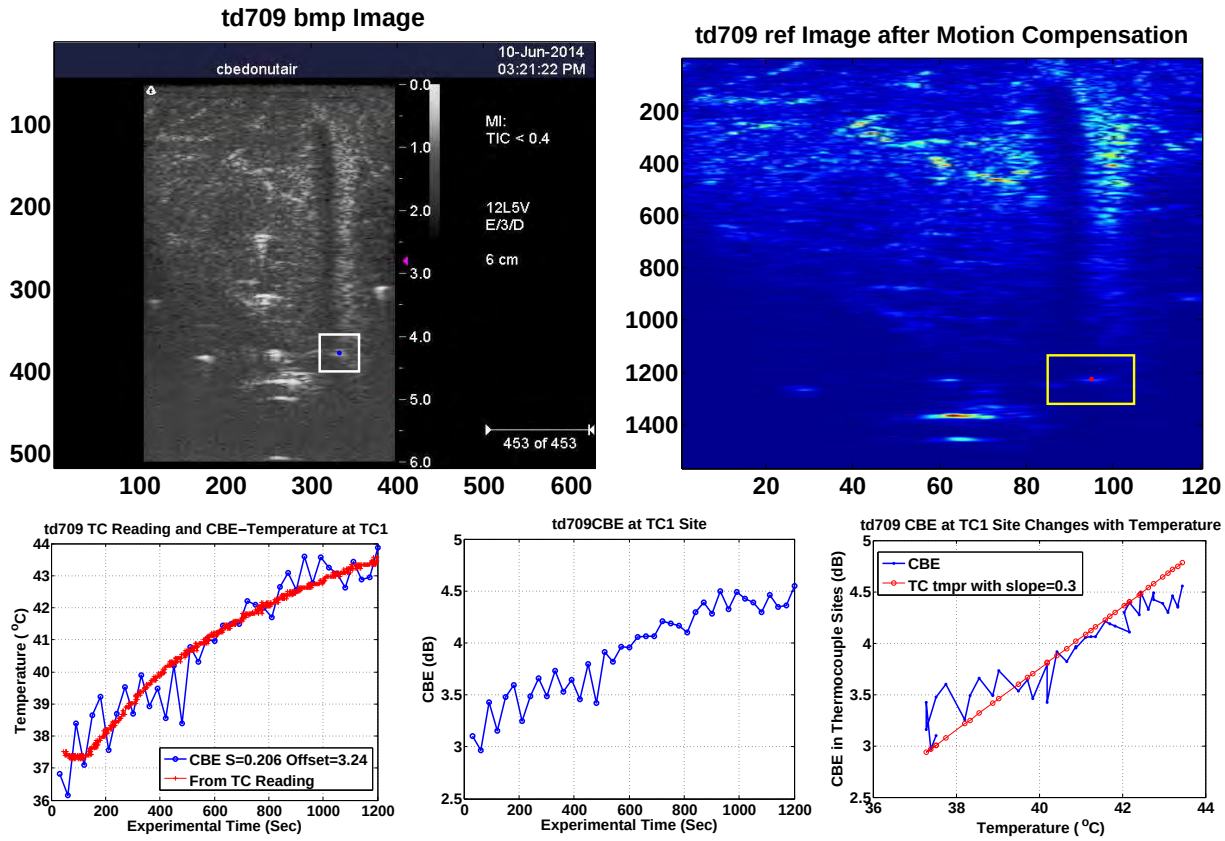


Figure 4.11: Temperature Sensitivity of CBE. Upper Left) Image with thermocouples in view, before experiment start. The white rectangle and blue dot identify the location of TC1. Upper Right) The reference image after motion compensation, with thermocouples pulled out of view. Yellow rectangle and red dot show the predicted location of TC1, based on it's location in the Terason gray-scale image. Lower Left) CBE estimated temperature at TC1 location as a function of time, along with the reading of TC1. Lower Center) Actual CBE change as a function of time at TC1 location. Lower Right) CBE at TC1 location as a function of TC1 temperature reading. Red line with slope = 0.3 serves as a reference [3].

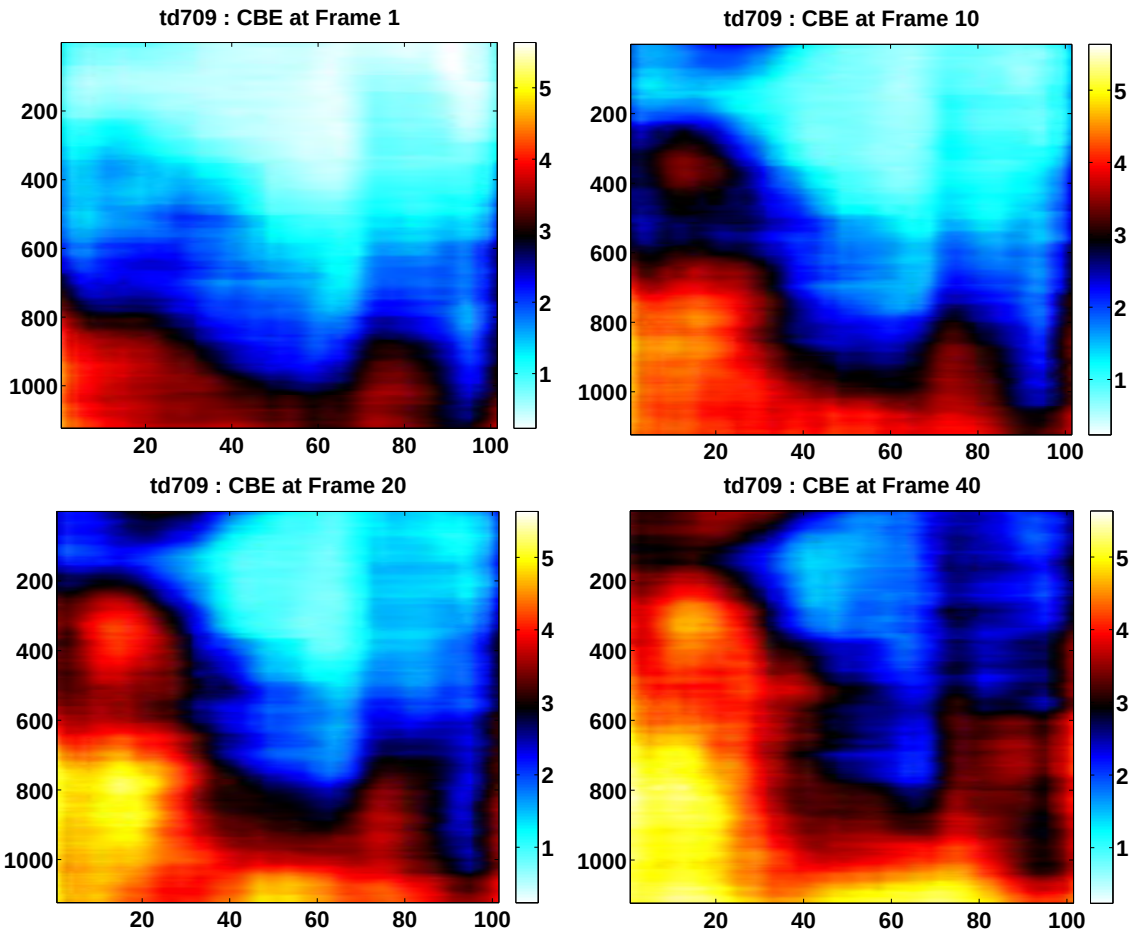


Figure 4.12: CBE images at frames # 1, 10, 20, 40. Color scale is in dB.

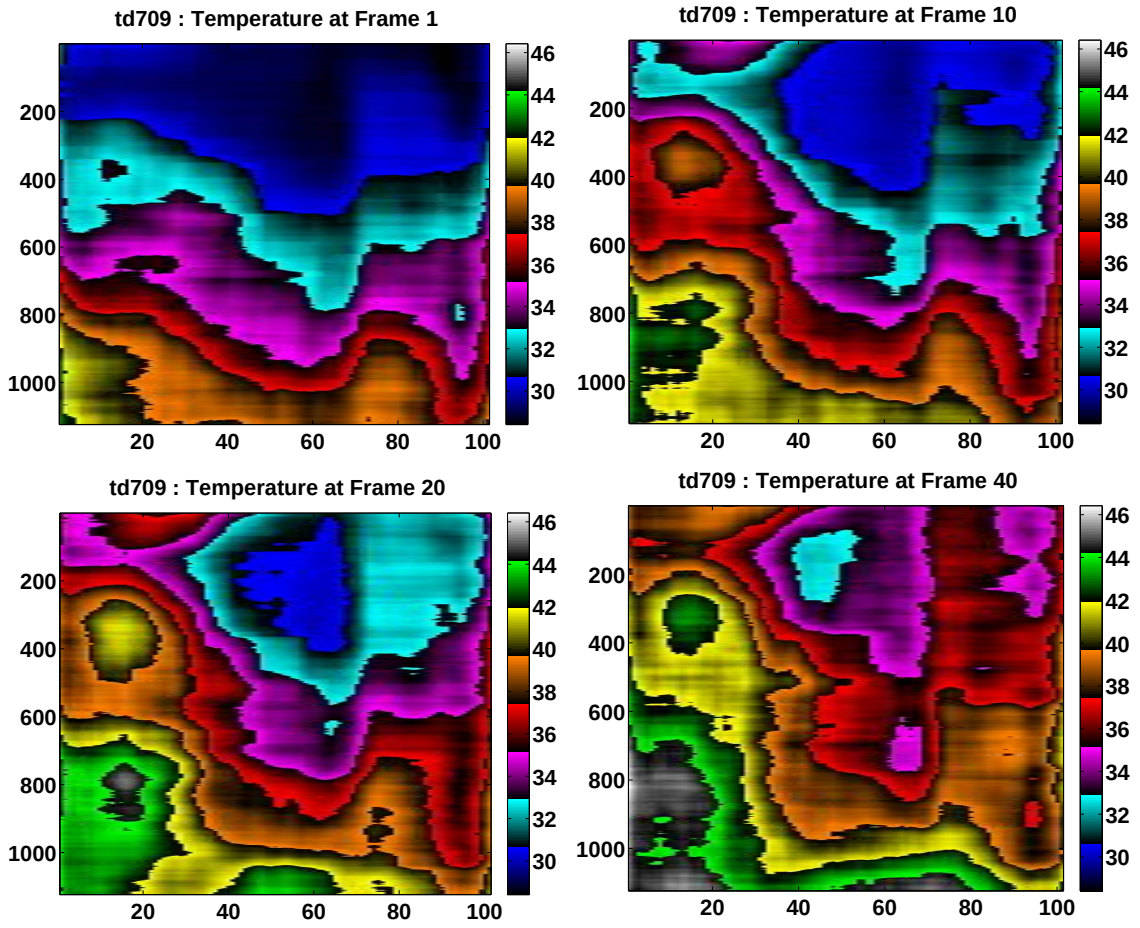


Figure 4.13: Temperature images at frames # 1, 10, 20, 40. Color scale is in degrees Celsius.

4.4.2 td710

From Fig. 4.14 we can see that the focus of the ultrasonic signal was at 3 cm, which gave us a relatively smaller region to work on. We can also tell that the B-mode ultrasonic images doesn't give us much information about temperature. Thus we need to analyze each experiment and find their CBE-Temperature relationship specifically.

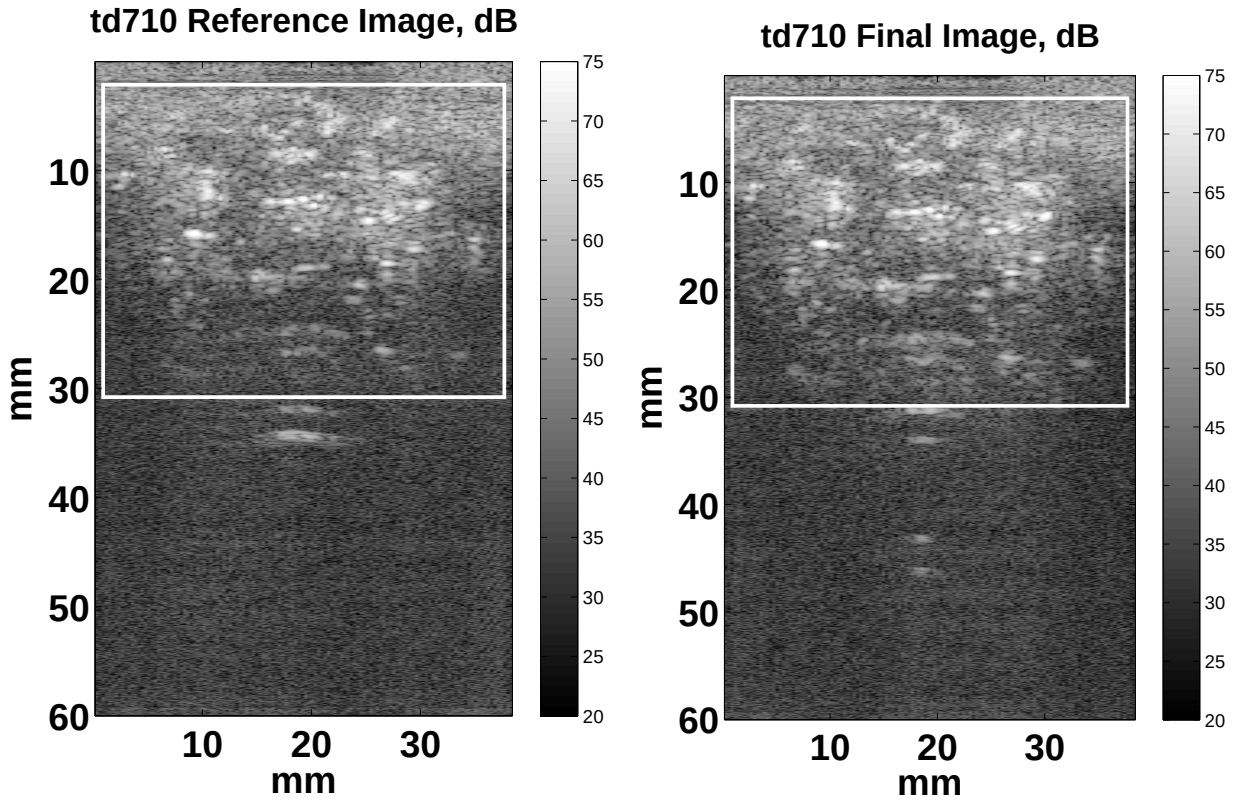


Figure 4.14: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image. Right) Last image of the experiment run (after 20 minutes).

Fig. 4.15 demonstrates the motion accumulation over experiment time at 25 spots in the ROI. The values were amplified by 50 for better visibility. Quiver motions were plotted on frame #1, 20, 30, 40.

The TC1 location was found based on its coordinates on Fig. 4.16 (Left). With adjustments of image shifts accounted for, the CBE value at TC1 site was calculated. As we can see on Fig. 4.16 (Center). Along with the CBE plotted against TC1 temperature readings, shown

on Fig. 4.16 (Right). The CBE-Temperature sensitivity was 0.0678 dB/^{o}C in td710 with an error of 1.1 $\pm$1.0^{o}C. The sensitivity was five times smaller than 0.3 dB/^{o}, which is the sensitivity we gained from previous the 3D experiments done by Basu [10].

We believe the low CBE-Temperature sensitivity was caused mainly by temperature gradient, since TC1 was close to hot-water tube during the non-uniform heating. Furthermore with only the 2D information, we can't compensation the motion in elevation direction, and that might cause error in CBE calculation. There was also a difference between the actual temperature at the TC1 spot in image acquisition view and the temperature readings of TC1 which was pulled out of the view before the heating started.

The region size (6x6 mm^2 in this case) selected for CBE calculation could be another cause. We want the region to be as large as possible to obtain a stable CBE value but also as small as possible to capture the temperature gradient.

6x6 mm^2, however, was too small to get enough valuable data based on our current method (energy averaging) of CBE calculation, while it's large enough to include many temperature gradient. We can get a huge improvement on accuracy in a smaller region size using ratio probability density function method [4] to calculate CBE, according to our previous work, ratio probability density function method can maintain a good accuracy even in a 0.02 cm^2 region. CBE and CBE estimated temperature of td710 at frames # 1, 10, 20, 40 were displayed in Fig. 4.17 and Fig. 4.18.

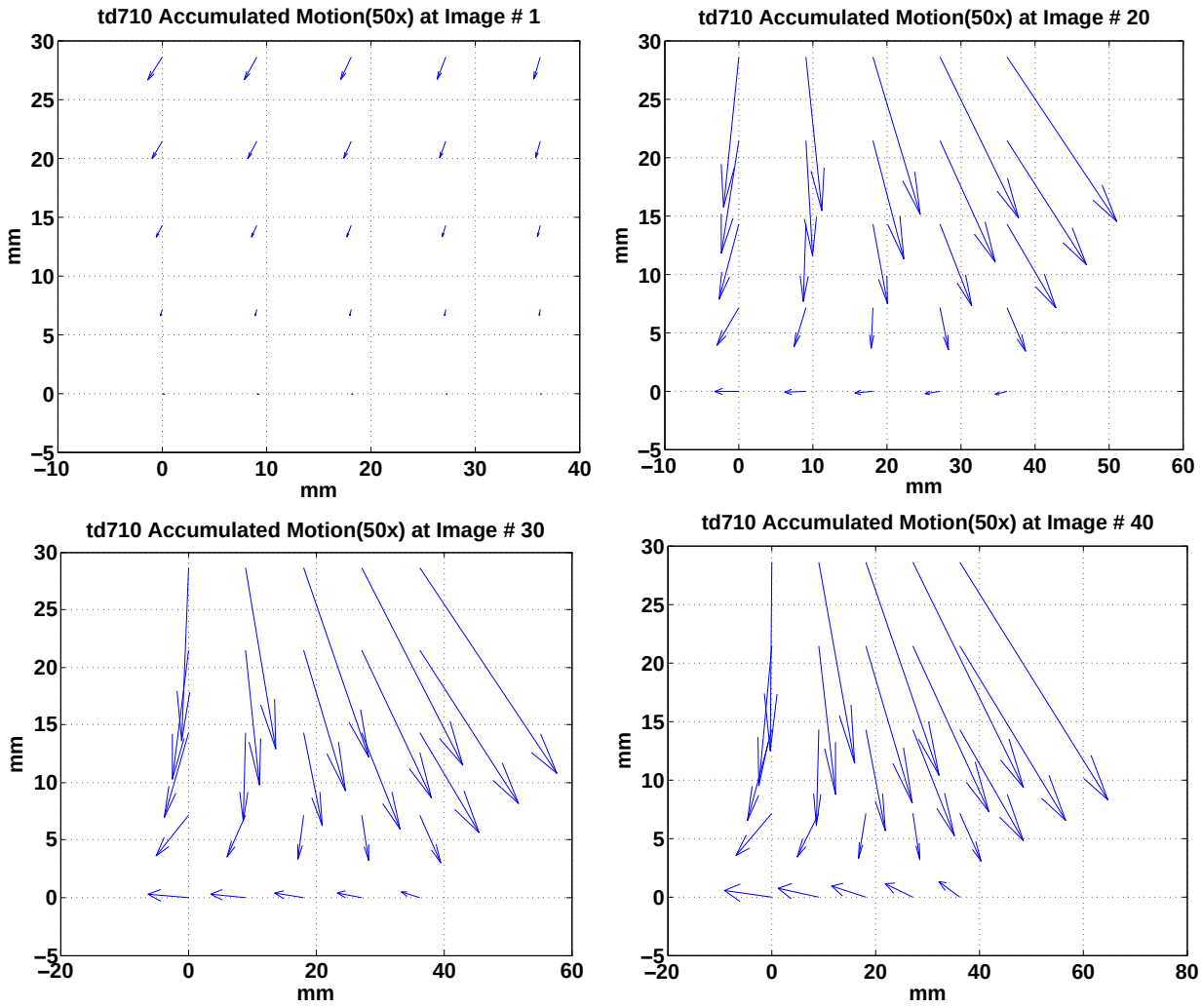


Figure 4.15: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

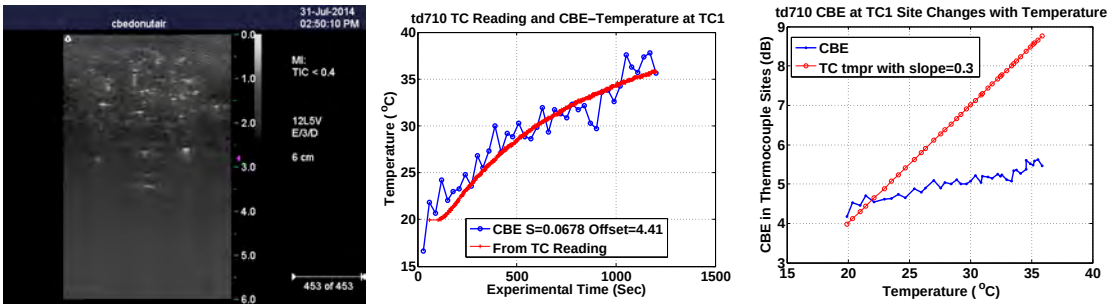


Figure 4.16: Left) Image with thermocouples in view, before experiment start. Center) Actual CBE change as a function of time at TC1 location. Right) CBE at TC1 location as a function of TC1 temperature reading. Red line with slope = 0.3 serves as a reference [3].

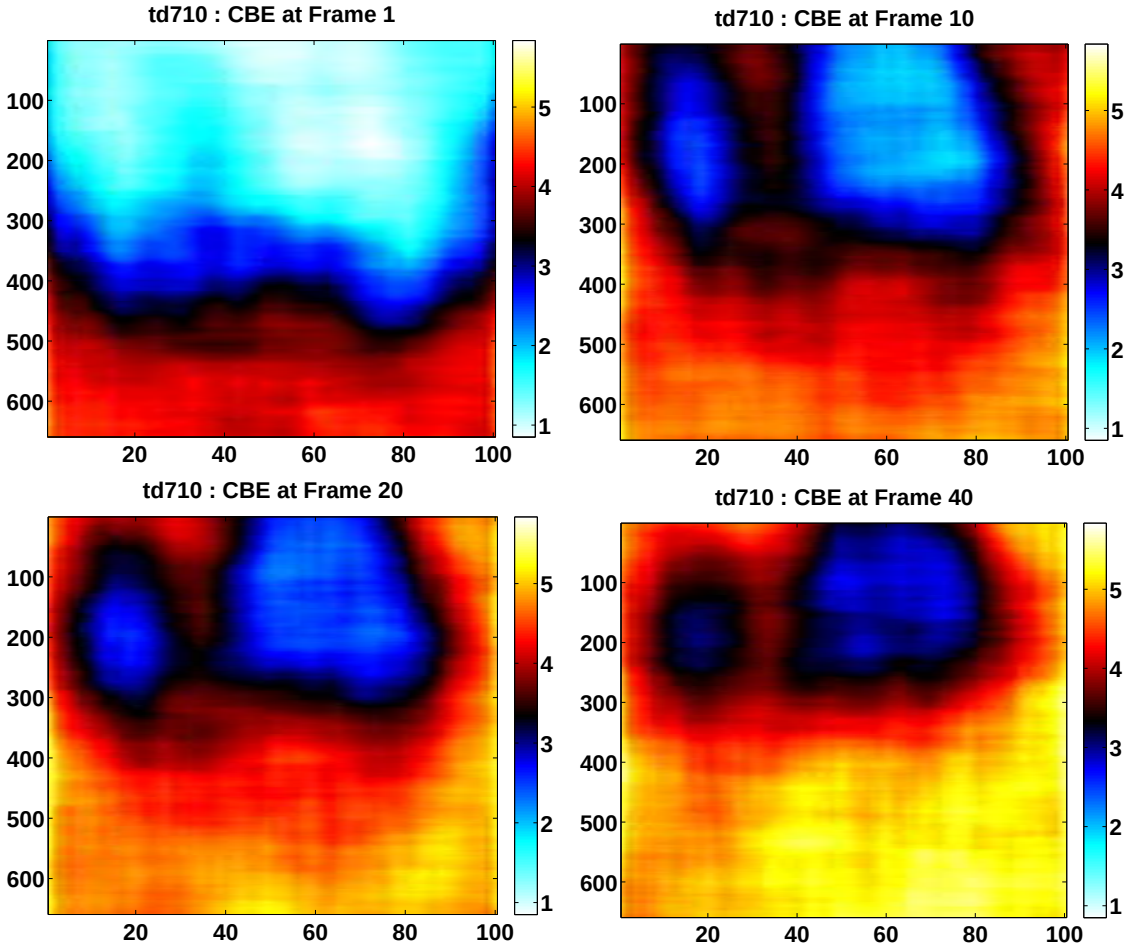


Figure 4.17: CBE images at frames # 1, 10, 20, 40. Color scale is in dB

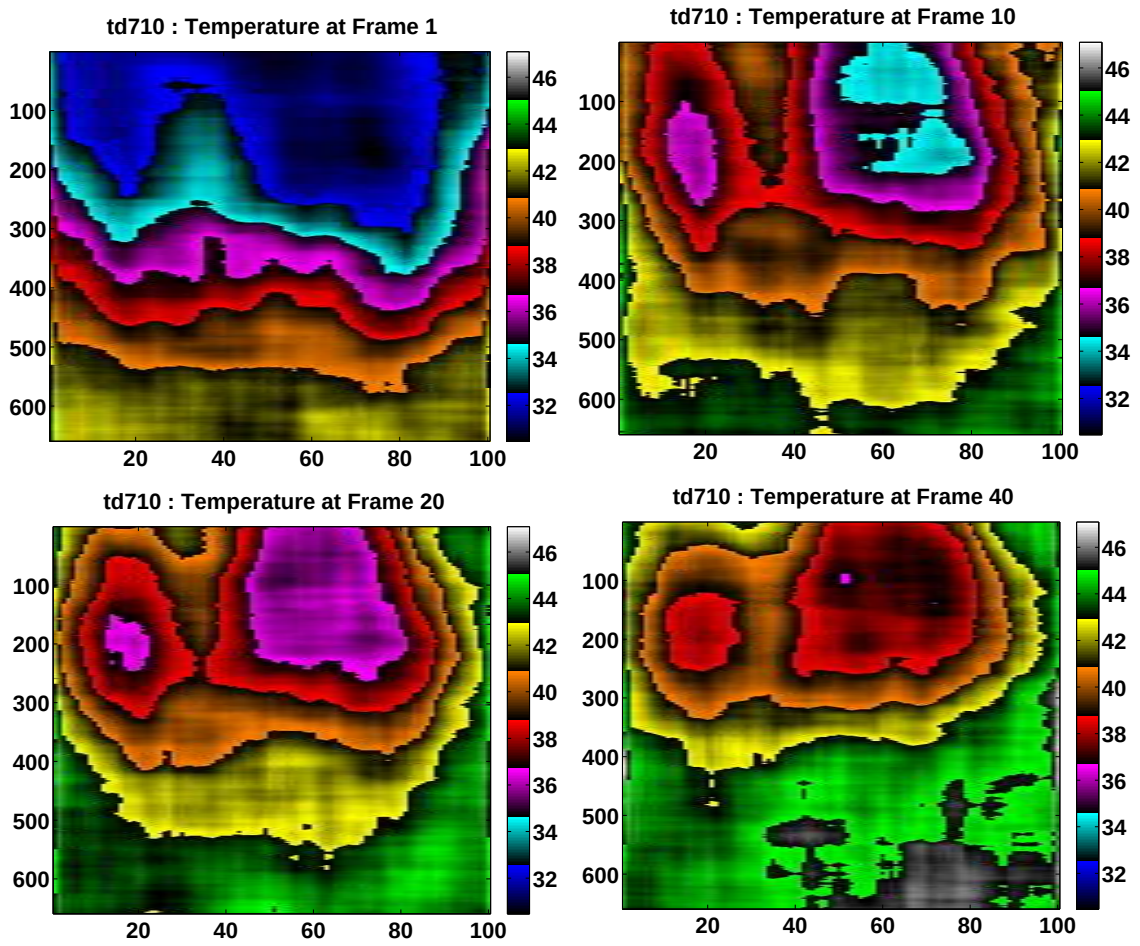


Figure 4.18: Temperature images at frames # 1, 10, 20, 40. Color scale is in degrees Celsius.

4.4.3 td714

Ultrasonic B-mode images of td714 reference image and last frame image were shown in Fig. 4.19. Motion in quiver was plotted at frames #1, 20, 30, 40, displayed in Fig. 4.20. CBE value at TC1 was calculated with a temperature sensitivity of 0.114 dB/^{o}C and error of 0.80 $\pm$ 0.58^{o}C as shown in Fig. 4.21. Fig. 4.23 demonstrates the temperature spread on frames #1, 10, 20, 40, with part of the water tube in the view.

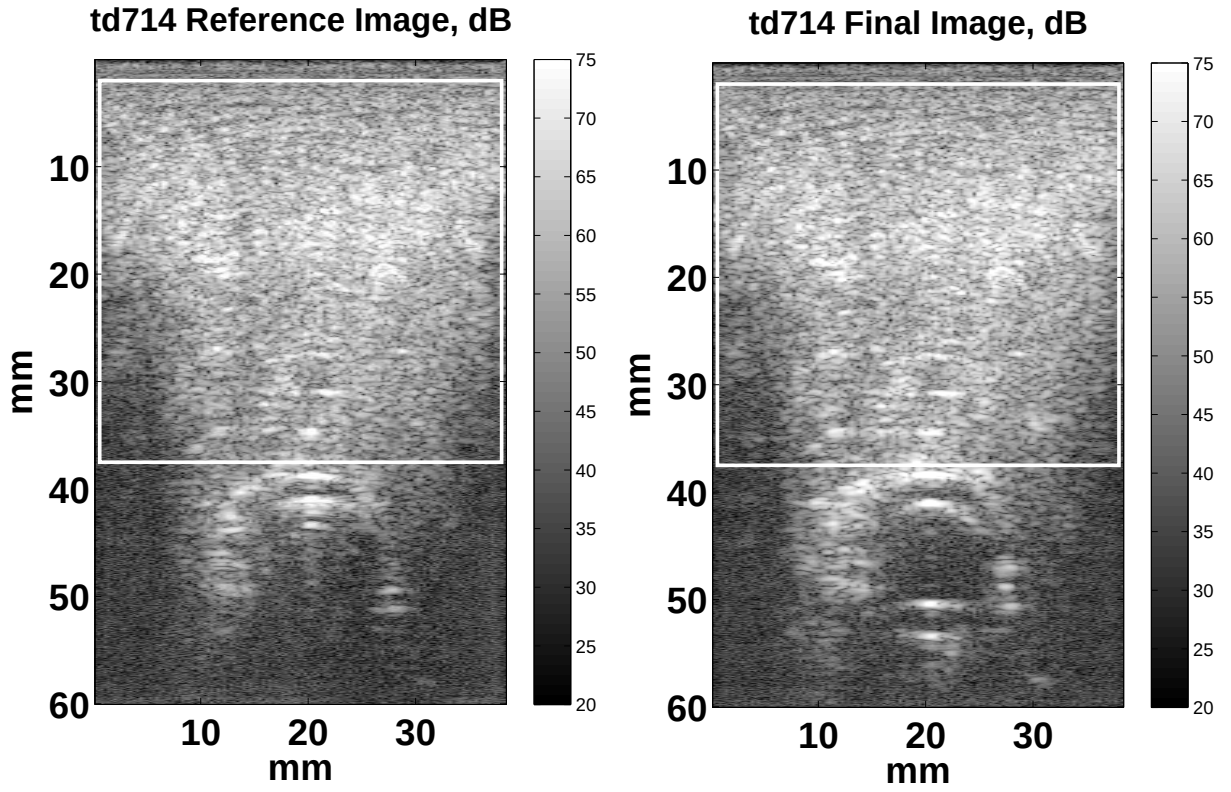


Figure 4.19: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image . Right) Last image of the experiment run (after 20 minutes).

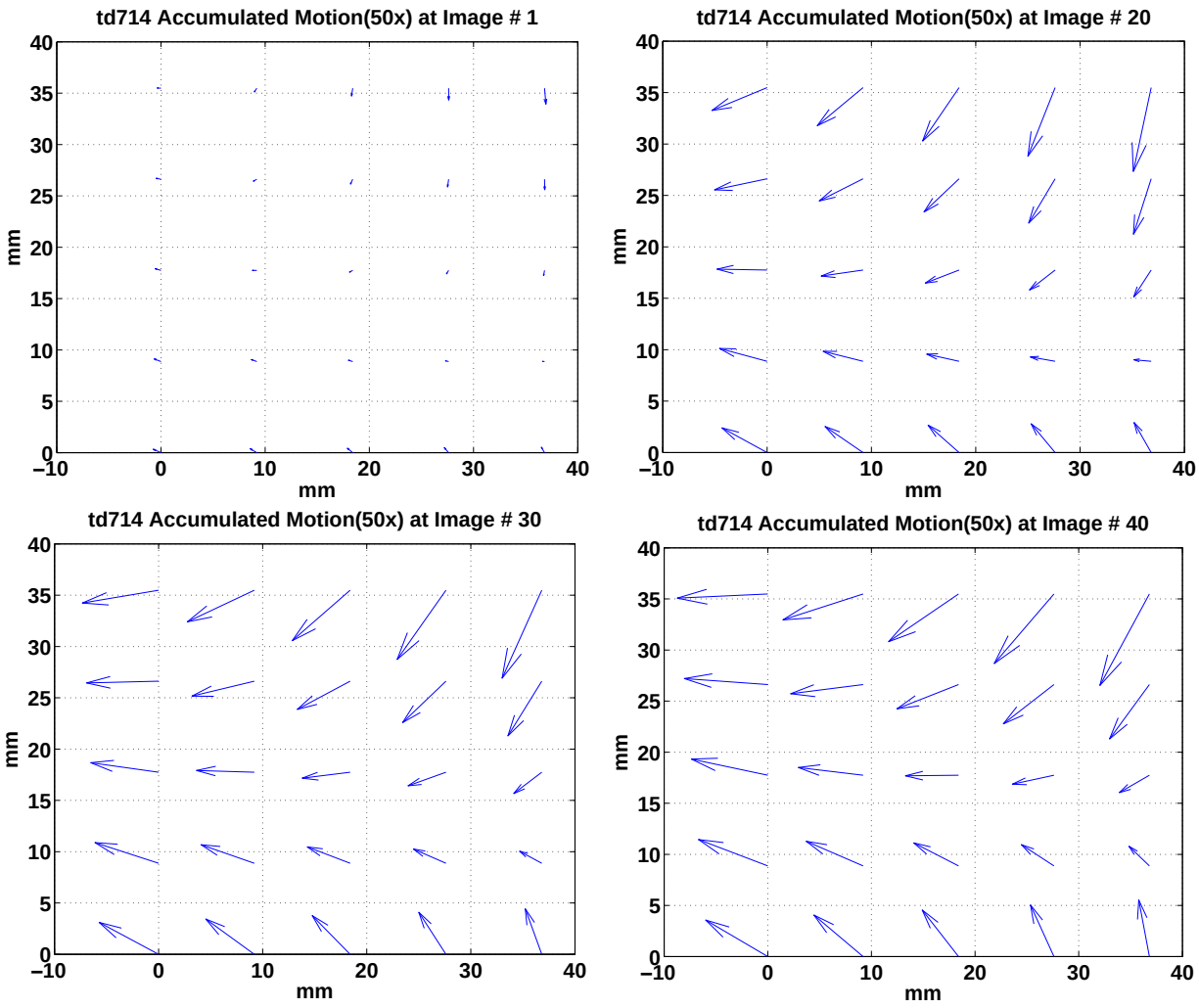


Figure 4.20: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

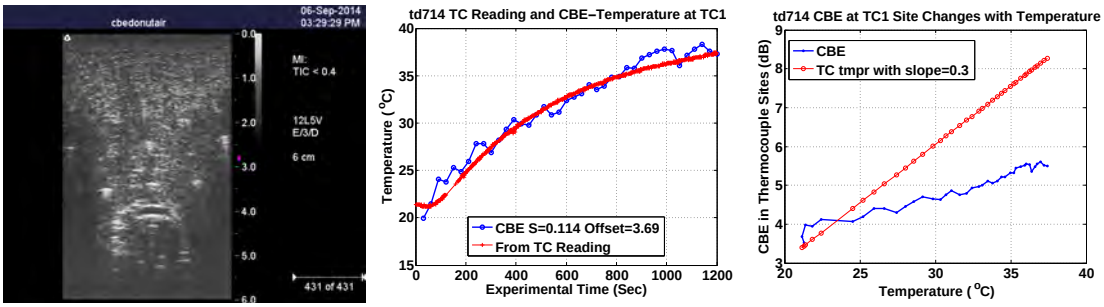


Figure 4.21: Left) CBE estimated temperature at TC1 location as a function of time, along with the reading of TC1. Center) Actual CBE change as a function of time at TC1 location. Right) CBE at TC1 location as a function of TC1 temperature reading. Red line with slope = 0.3 serves as a reference [3].

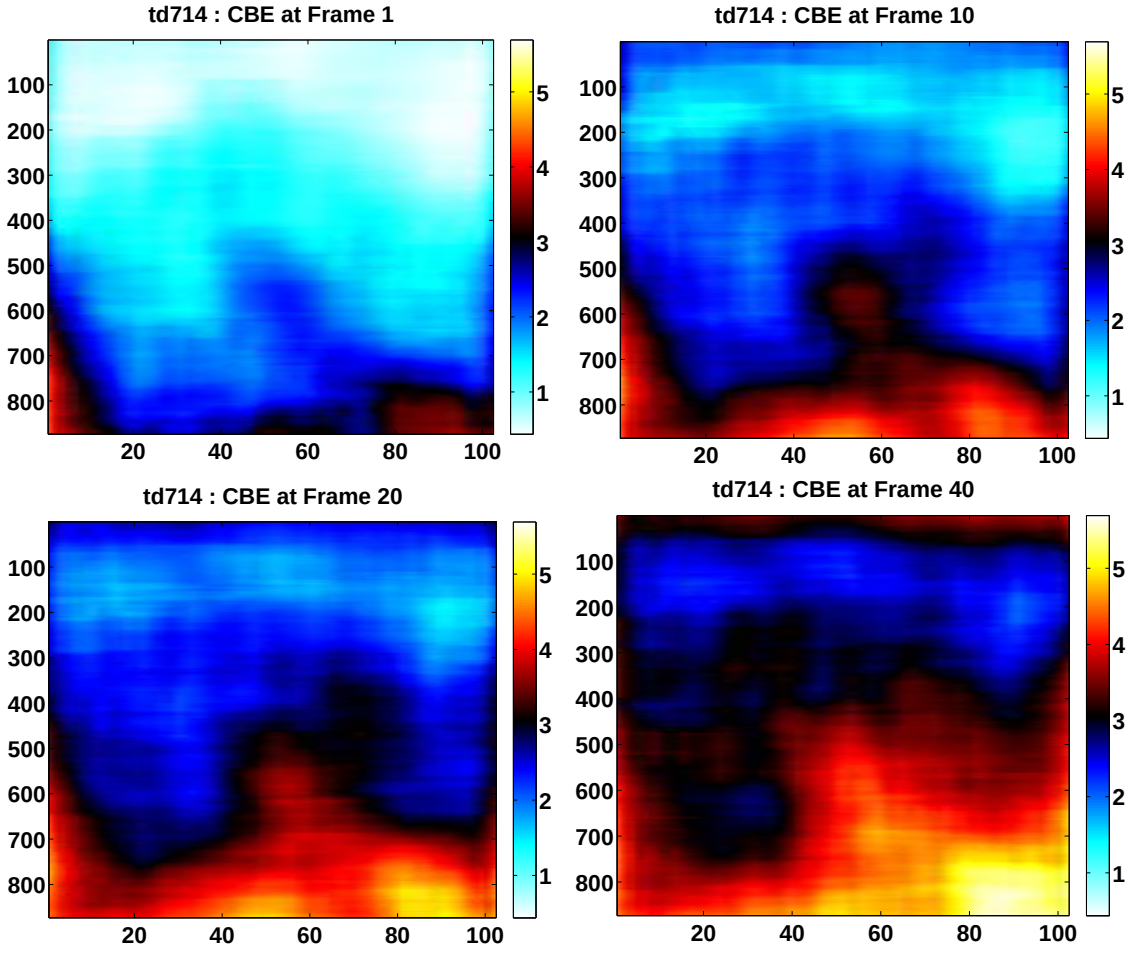


Figure 4.22: CBE images at frames # 1, 10, 20, 40. Color scale is in dB

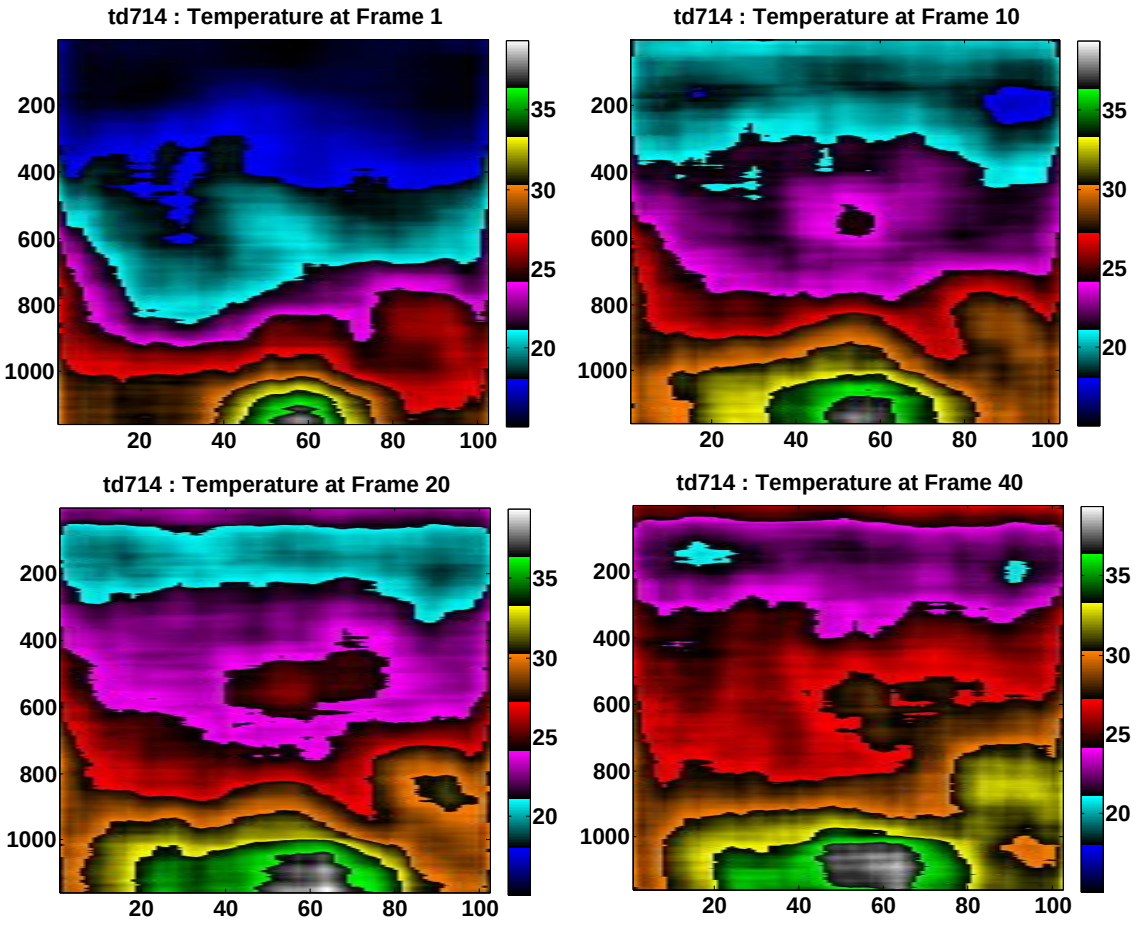


Figure 4.23: Temperature images at frames # 1, 10, 20, 40. Color scale is in degrees Celsius. With water tube in the view.

4.5 Less than Ideal Experiments: td711, td712 and td713

Td711 was categorized in Less than Ideal Experiments because a one-month-old tissue was used as specimen. For td713, the turkey specimen was dehydrated for 12 hours prior the experiment. In both cases, we didn't have a good ultrasonic penetration through the tissue. For td712, the correlation coefficient between reference image and other images after MC (0.77) was lower than any other experiments (greater than 0.9), which suggests a data acquisition error. Even though they were not ideal either in experimental setup or data acquisition, they are still good lesson-teaching experience.

In this section, we also studied the CBE-Temperature relationship for each experiment. For td711 and td713 we didn't use TC1 because it's not in the ultrasonic B-mode image view before the heating started as seen in Fig. 4.26 (left) and Fig. 4.34(left), hence we don't have the basis to calculate the TC1 coordinates in motion-compensated images. We picked TC8 instead for td711 and td713 because TC3, TC5 were broken, and TC4, TC7 were too close to ROI edge to have a 6x6 mm^2 region centered at them for CBE calculation.

Turns out TC8 has a good CBE estimation sensitivity for temperature. For td711, it's 0.284 dB/^{o}C with an error of 0.48 $\pm$ 0.39^{o}C in temperature estimation. For td713, 0.336 dB/^{o}C with an error of 1.0 $\pm$ 0.68^{o}C. We continued to use TC1 for td712, and got a CBE-Temperature sensitivity of 0.049^{o}C, the error was 1.8 $\pm$ 0.98^{o}C.

The results from estimating temperature with CBE at TC8 sites proved to be very promising compare to that at TC1. Thus we did the CBE-Temperature sensitivity analysis on all the other experiments using TC8 as well. The results are listed in Tab. 4.1. TC8 from td709 was not used, because it's reading seems erroneous according to Fig.4.7. We also included TC5 from td710 for the CBE-Temperature study, since it's the only experiments that TC5 was working properly.

4.5.1 td711

Ultrasonic B-mode images of td711 reference image and last frame image were shown in Fig. 4.24. Motion in quiver was plotted at frames #1, 20, 30, 40, displayed in Fig. 4.25. CBE value at TC8 was calculated with a temperature sensitivity of 0.28 dB/^{o}C as shown in Fig. 4.26. CBE of td711 at frames # 1, 10, 20, 40 were displayed in Fig. 4.27.

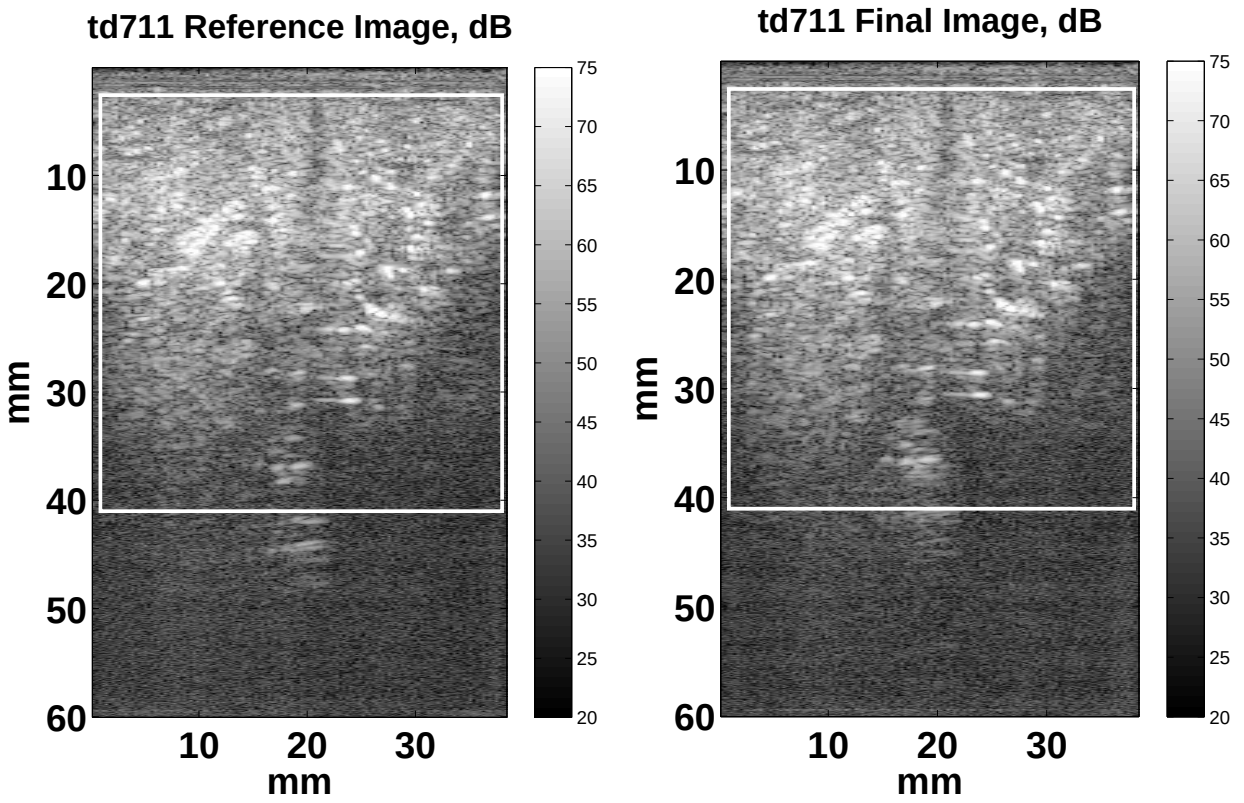


Figure 4.24: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image . Right) Last image of the experiment run (after 20 minutes).

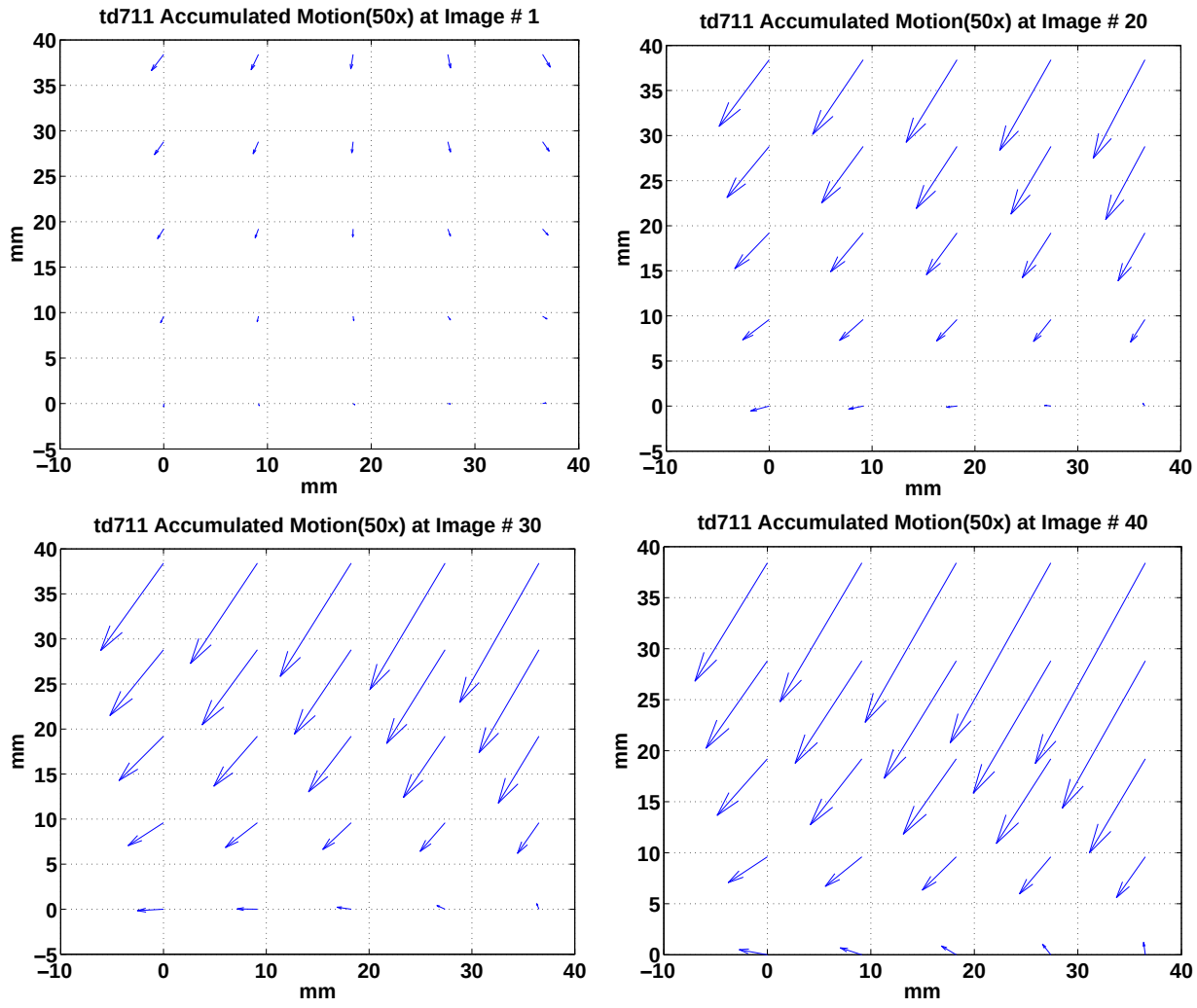


Figure 4.25: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

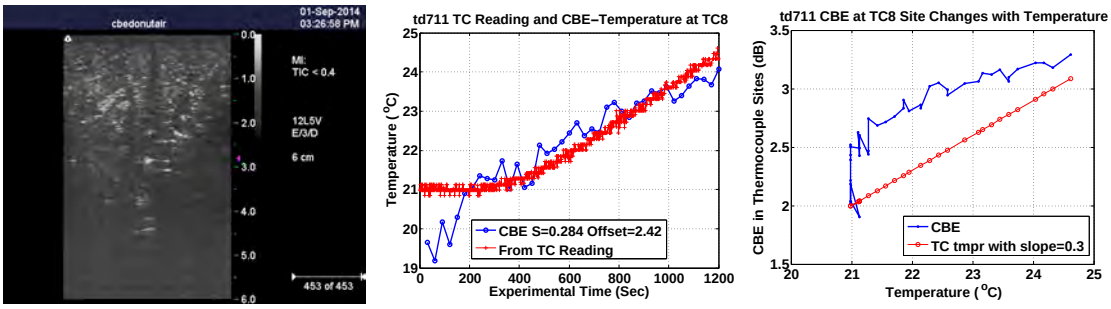


Figure 4.26: Left) CBE estimated temperature at TC8 location as a function of time, along with the reading of TC8. Center) Actual CBE change as a function of time at TC8 location. Right) CBE at TC8 location as a function of TC8 temperature reading. Red line with slope = 0.3 serves as a reference [3].

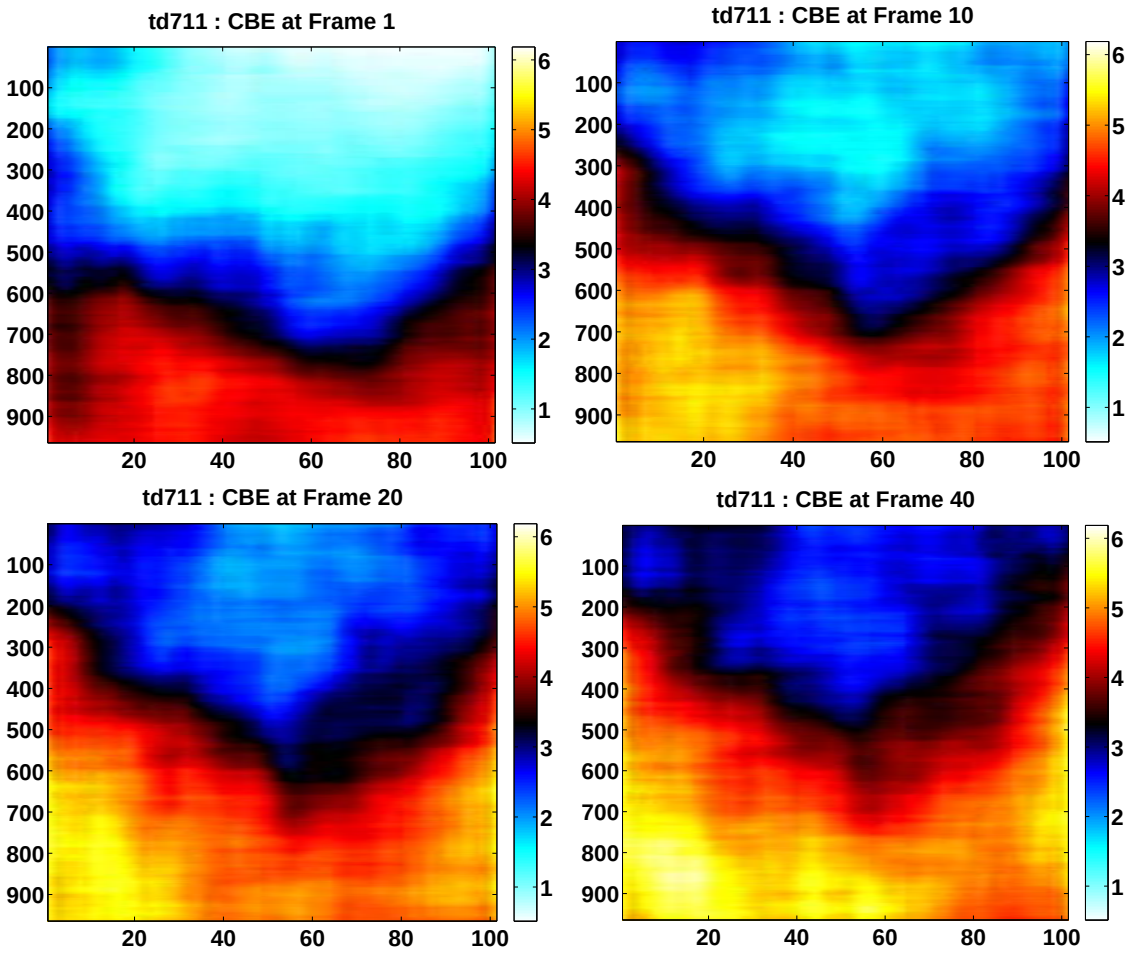


Figure 4.27: CBE images at frames # 1, 10, 20, 40. Color scale is in dB

4.5.2 td712

Ultrasonic B-mode images of td711 reference image and last frame image were shown in Fig. 4.28. Motion in quiver was plotted at frames #1, 20, 30, 40, displayed in Fig. 4.29. CBE value at TC1 was calculated with a temperature sensitivity of 0.050 dB/^{o}C as shown in Fig. 4.30. CBE of td712 at frames # 1, 10, 20, 40 were displayed in Fig. 4.27.

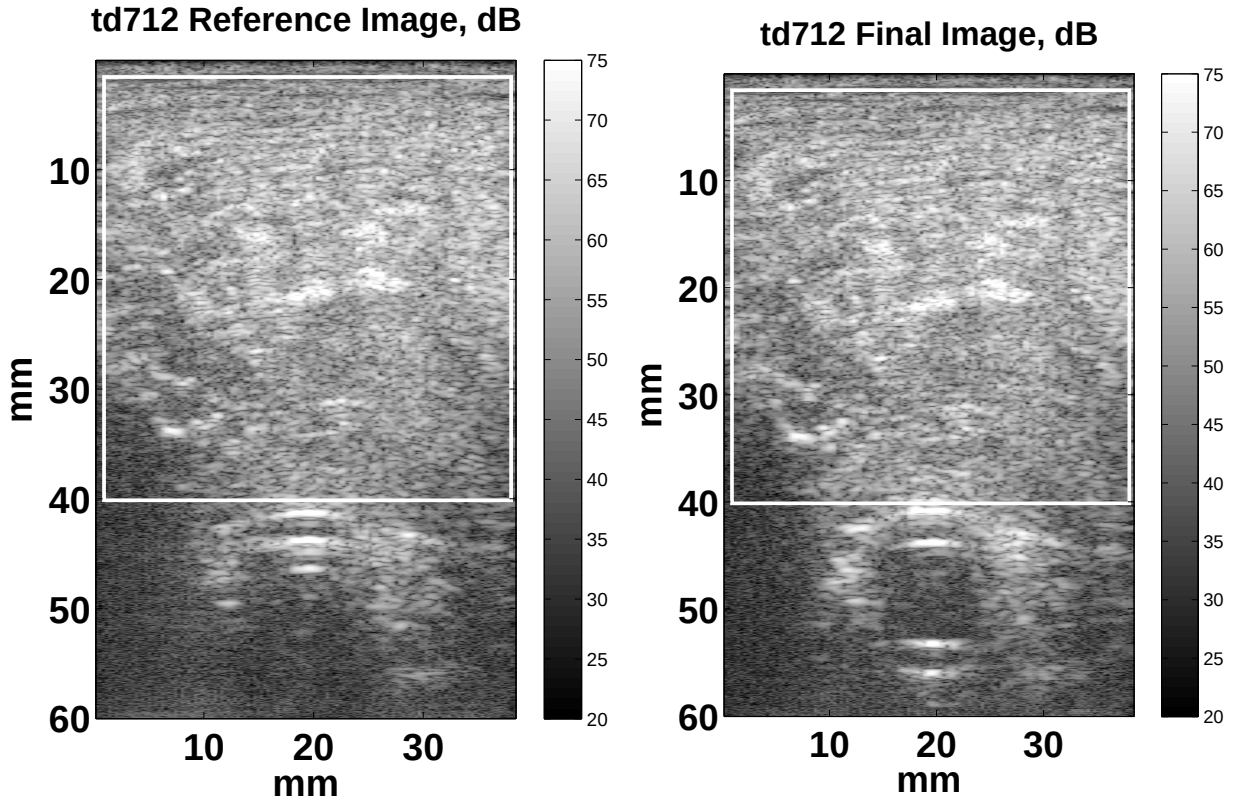


Figure 4.28: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image . Right) Last image of the experiment run (after 20 minutes).

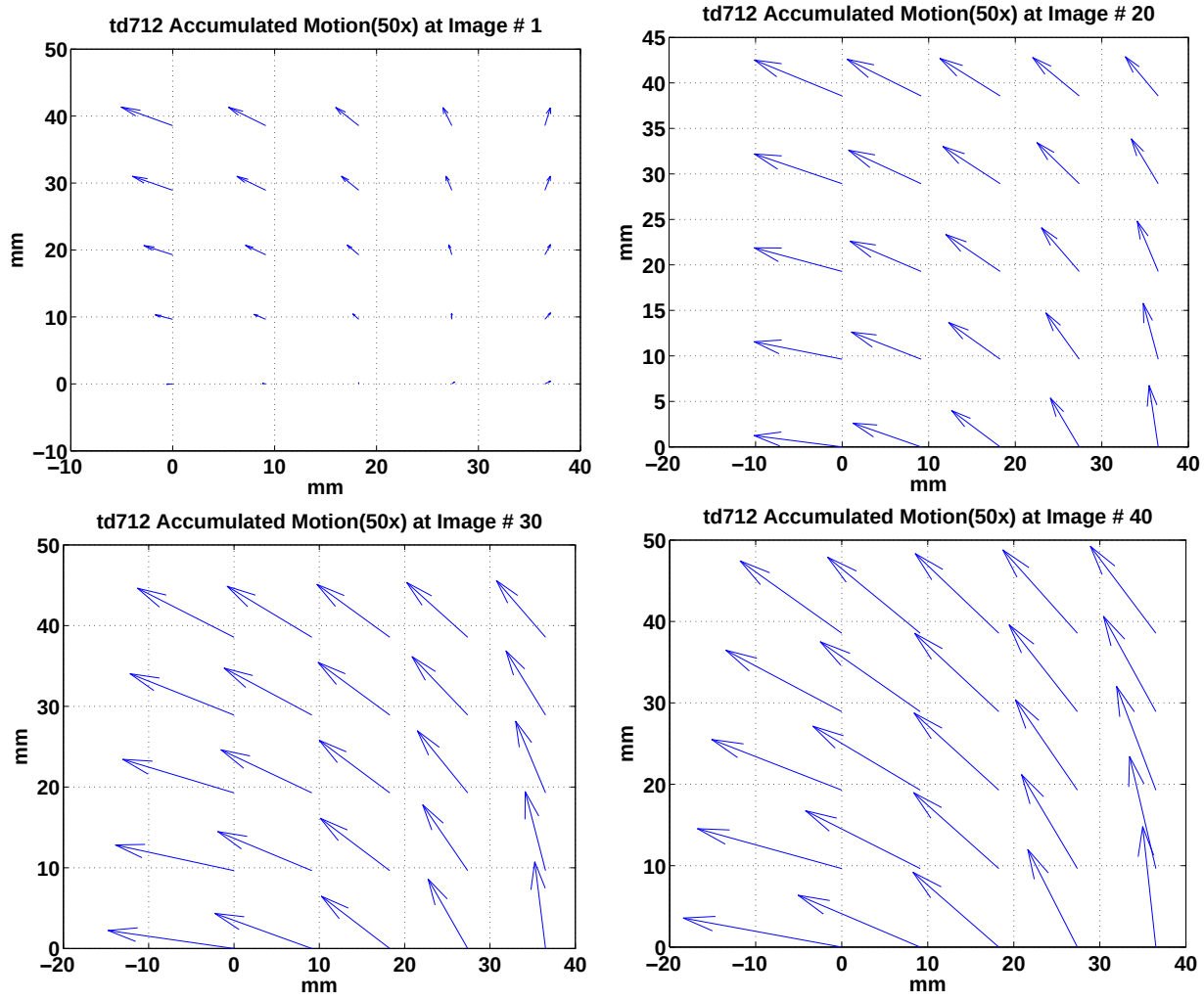


Figure 4.29: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

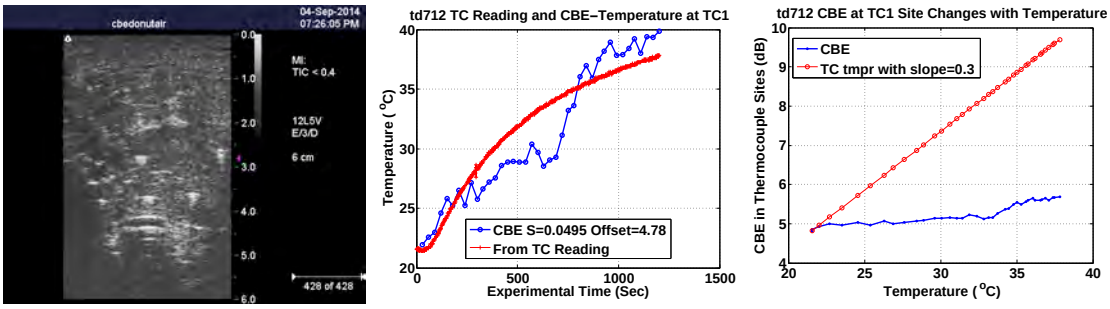


Figure 4.30: Left) CBE estimated temperature at TC1 location as a function of time, along with the reading of TC1. Center) Actual CBE change as a function of time at TC1 location. Right) CBE at TC1 location as a function of TC1 temperature reading. Red line with slope = 0.3 serves as a reference [3].

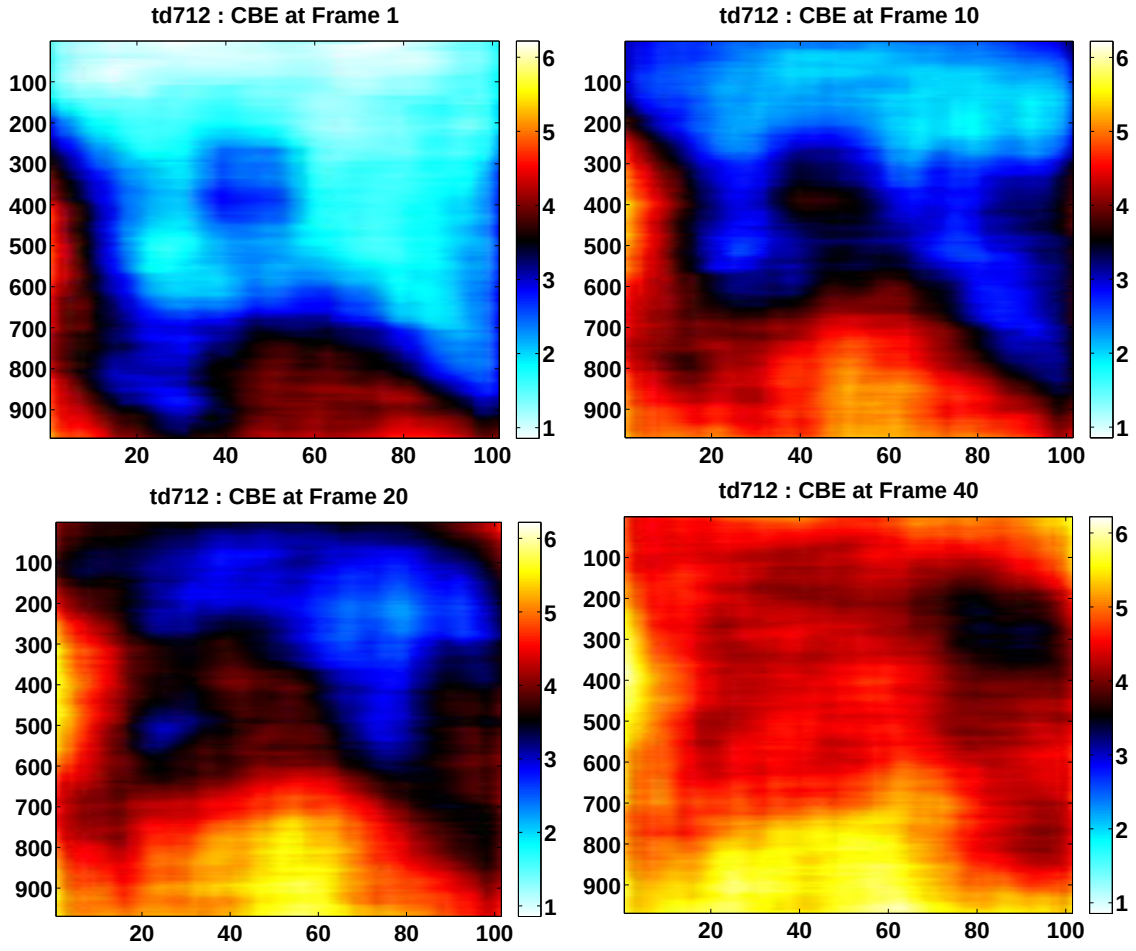


Figure 4.31: CBE images at frames # 1, 10, 20, 40. Color scale is in dB

4.5.3 td713

Ultrasonic B-mode images of td711 reference image and last frame image were shown in Fig. 4.32. Motion in quiver was plotted at frames #1, 20, 30, 40, displayed in Fig. 4.33. CBE value at TC8 was calculated with a temperature sensitivity of 0.33 dB/°C as shown in Fig. 4.34. CBE of td713 at frames # 1, 10, 20, 40 were displayed in Fig. 4.27.

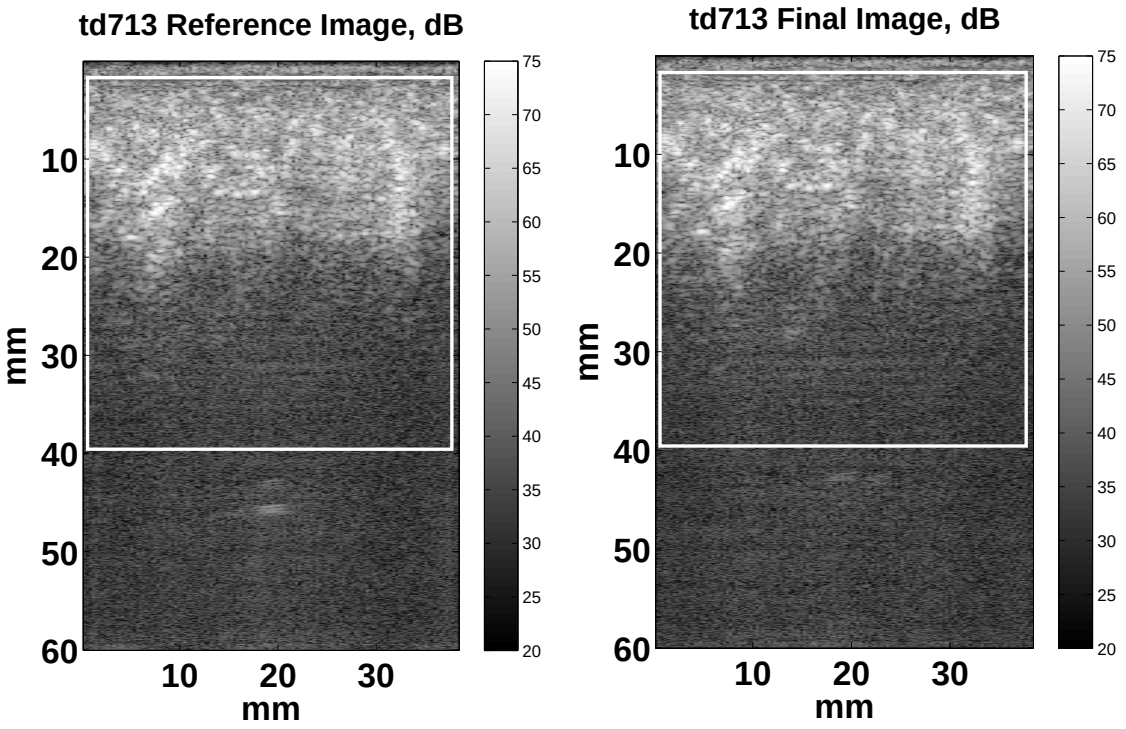


Figure 4.32: B-mode ultrasonic images in which the white rectangle indicates the ROI. Left) Reference image . Right) Last image of the experiment run (after 20 minutes).

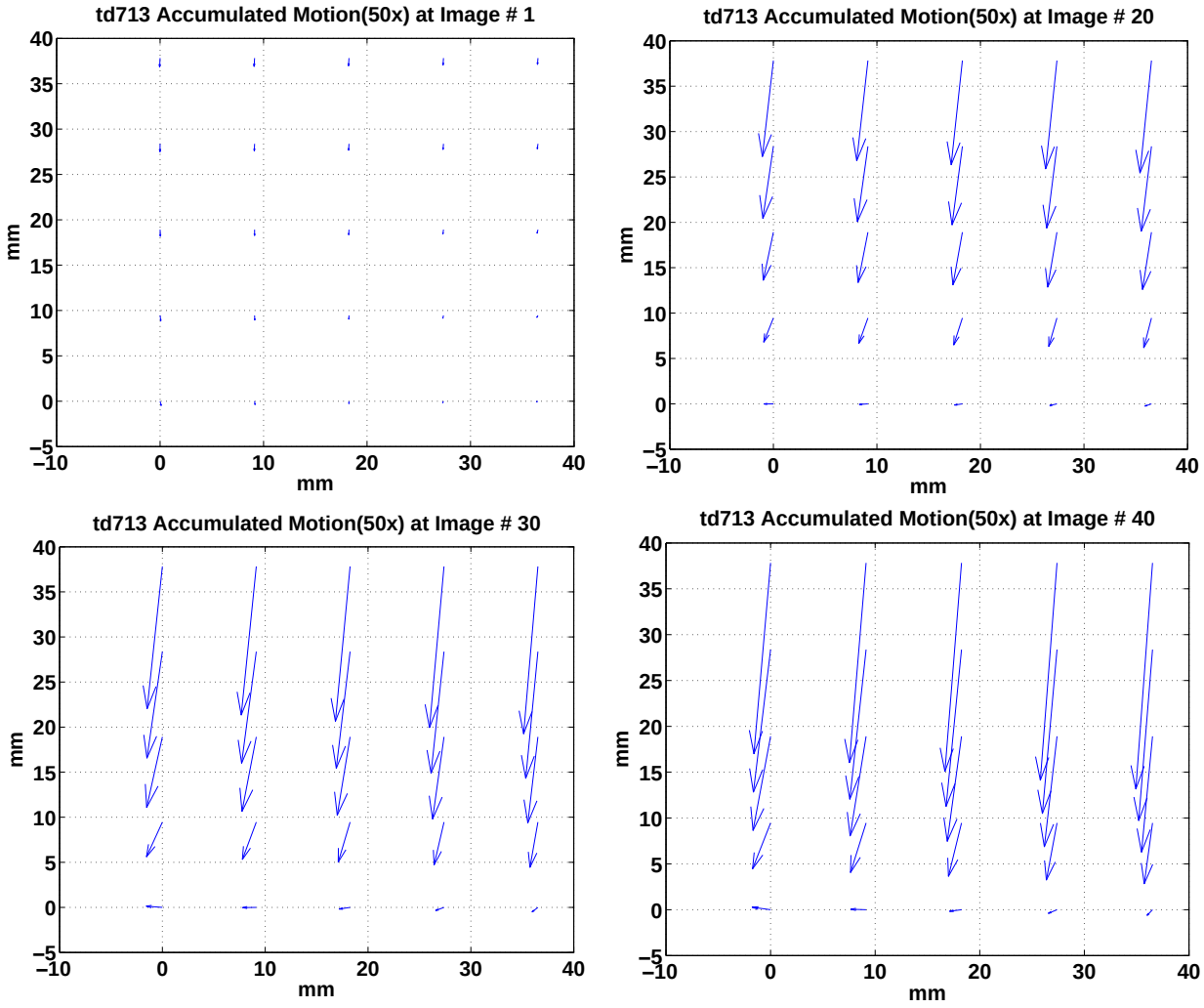


Figure 4.33: Accumulated motion in quiver images # 1, 20, 30, 40. The quiver display demonstrates both the direction and the value of motion at 25 distinct spots in ROI. Motion amplitude is amplified by a factor of 50 for visibility.

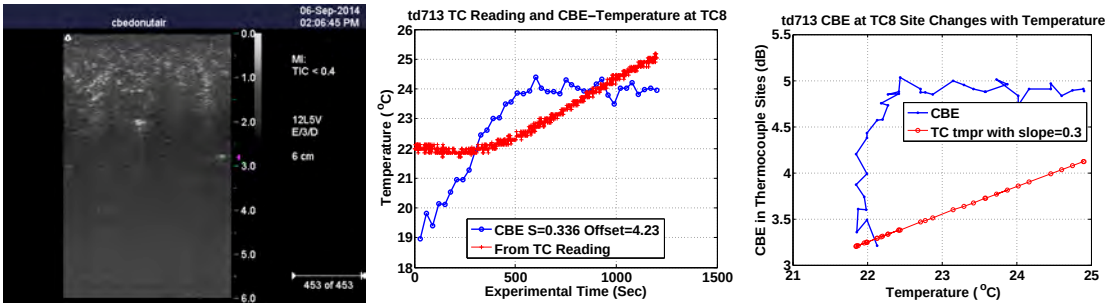


Figure 4.34: Left) CBE estimated temperature at TC8 location as a function of time, along with the reading of TC8. Center) Actual CBE change as a function of time at TC8 location. Right) CBE at TC8 location as a function of TC8 temperature reading. Red line with slope = 0.3 serves as a reference [3].

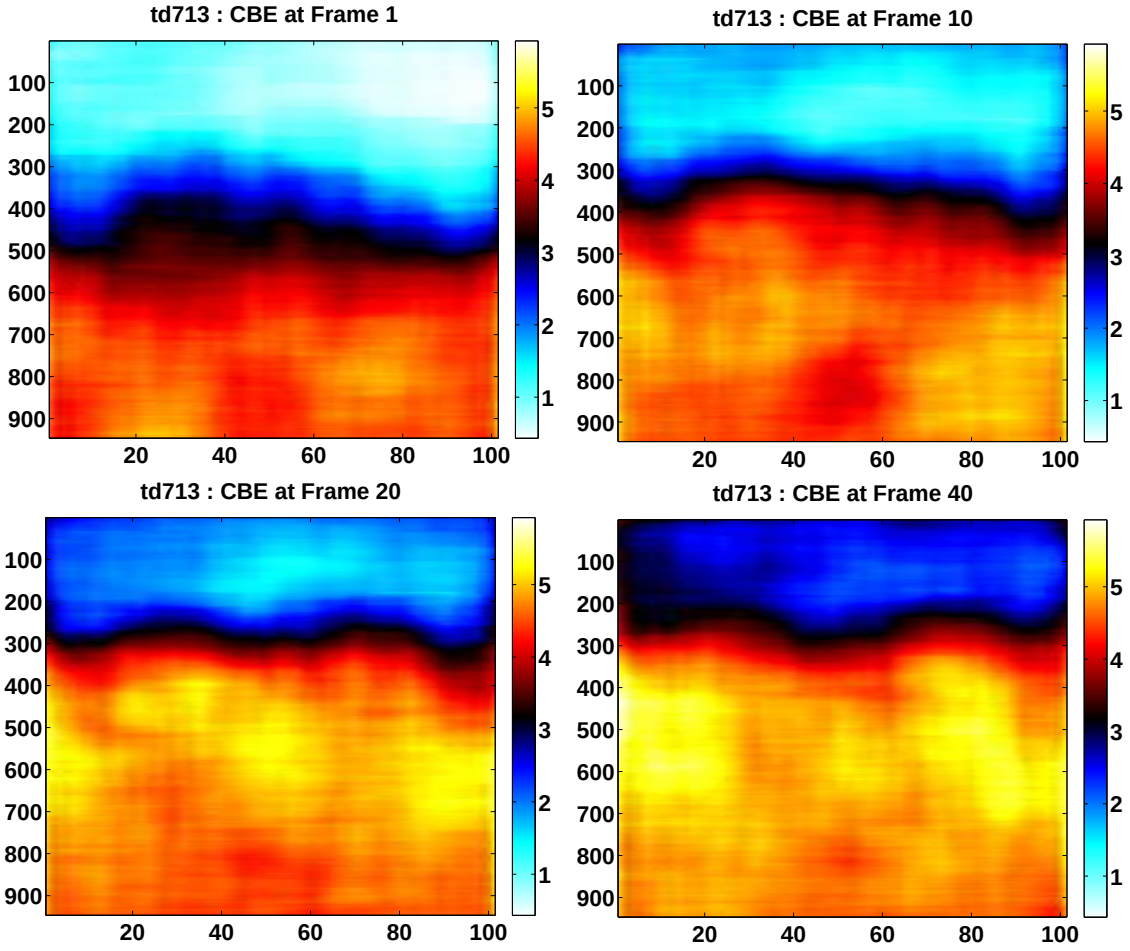


Figure 4.35: CBE images at frames # 1, 10, 20, 40. Color scale is in dB

4.6 Summary

From the previous 3D study done by our group, the temperature sensitivity of CBE was 0.3 dB/^{o}C on average[10]. We wanted to see if this sensitivity still holds in our 2D temperature imaging system. From the data in Tab. 4.1, it's clear that the average CBE temperature sensitivity comes from the calculation based on TC8 was equal to 0.31 dB/^{o}C, with an error of 0.61$\pm$0.66^{o}C. The calculations based on TC1, however, demonstrated a 3 times smaller sensitivity, with a 1.5 times larger error compare to TC8.

At first we thought TC1 would be the best candidate for CBE-Temperature study, because it's close enough to heat source to be exposed to a large thermal energy, meanwhile far enough that the interference of temperature gradient from hot-water tube is not as big as TC2. The results, however, told us another story: the further the thermocouple from the heat source, the closer the CBE-Temperature sensitivity to 0.3 dB/^{o}C and the better the estimation accuracy. We can see from Tab. 4.1 that the average CBE-Temperature sensitivity over 10 thermocouple sites is 0.21 $\pm$ 0.11 dB/^{o}C, with an error of 0.78 $\pm$ 0.81 dB/^{o}C.

We think the reasons lie at three areas as we discussed about earlier in this chapter: temperature gradient near hot-water tube, lack of information in elevation direction, and the poor performance of our current CBE calculation method in a 6x6 mm^2 region. Further investigation will be carried out in the future to find out the influence of the above three factors too the CBE-Temperature estimation sensitivity and accuracy.

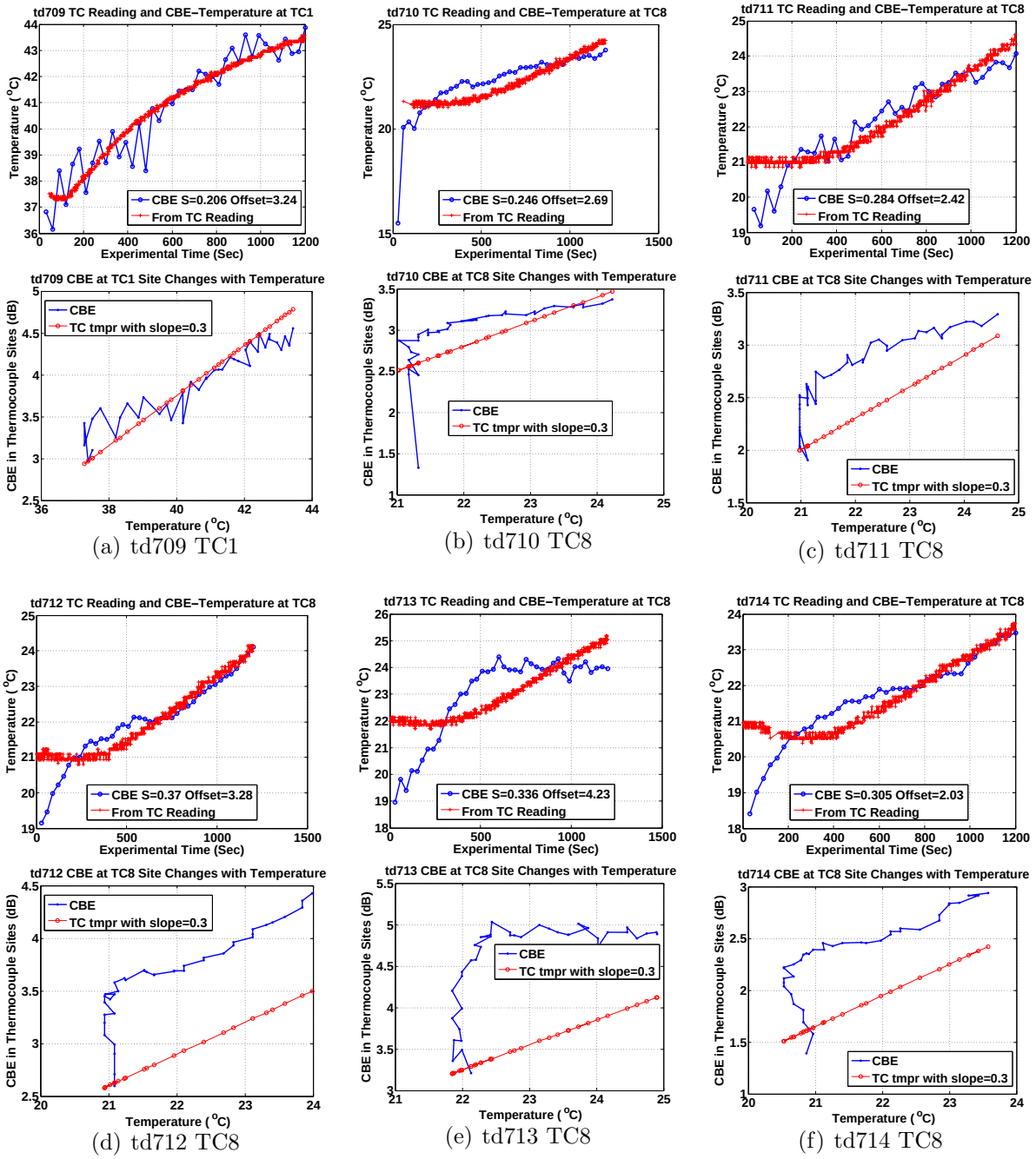
Fig. 4.37 shows the temperature estimated based on CBE at TC1 site for all experiments where TC1 was visible in image acquisition view before experiment started. We can tell that the CBE-Temperature sensitivity obtained based on TC1 was not ideal (not close to 0.3 dB/^{o}C), except for the one in td709 (0.21 dB/^{o}C). We think it's because TC1 in td709 was left in the image acquisition view during the entire experiment, which would provide the exact temperature reading at the TC1 site in the view. This observation made us reconsider the effect of a thermocouple tip left in the view of ultrasonic image acquisition to the accuracy of the raw ultrasonic image data acquisition. Further investigation is needed to have a good answer.

On the other hand, Fig. 4.36 includes the best usable thermocouples for temperature estimation with CBE in each experiment. Fig. 4.37 shows the CBE estimated temperature at TC1 site. From the CBE temperature sensitivity in each site we can see that the best location among TC1, TC5 and TC8 is TC8.

Tab. 4.1 demonstrated the CBE-Temperature sensitivity from calculation over 10 usable thermocouple sites in 6 experiment, along with their temperature estimation error. we can see that the average temperature sensitivity of CBE calculated based on the 5 TC8 sites was 0.31±0.045 dB/^{o}C , which very close to 0.3 dB/^{o}C, with an error of 0.61±0.66^{o}C in temperature estimation. Whereas, for the 4 TC1 sites, the average sensitivity was 0.11 ±0.071 dB/^{o}C, which is 3 times smaller than the expected CBE temperature sensitivity, and it's error was 1.08±0.93^{o}C which is 1.5 times larger than that of TC8 sites. We have the average sensitivity over 10 sites at 0.21±0.11 dB/^{o}C, and error of 0.78±0.81^{o}C.

Similar to Fig. 2.6, which illustrates the estimated temperature versus thermocouple readings at 7 locations from the 3D ultrasonic volume study done by Basu [10]. The accuracy of CBE estimated temperature at 10 thermocouple sites of our 2D study is also demonstrated in Fig. 4.39. Fig. 4.38 contains the corresponding temperature plots at 10 thermocouple sites, including the CBE estimated temperature at different CBE-Temperature sensitivities (0.3 dB/^{o}C and the slope value obtained from first-degree polynomial fit), along with thermocouple reading.

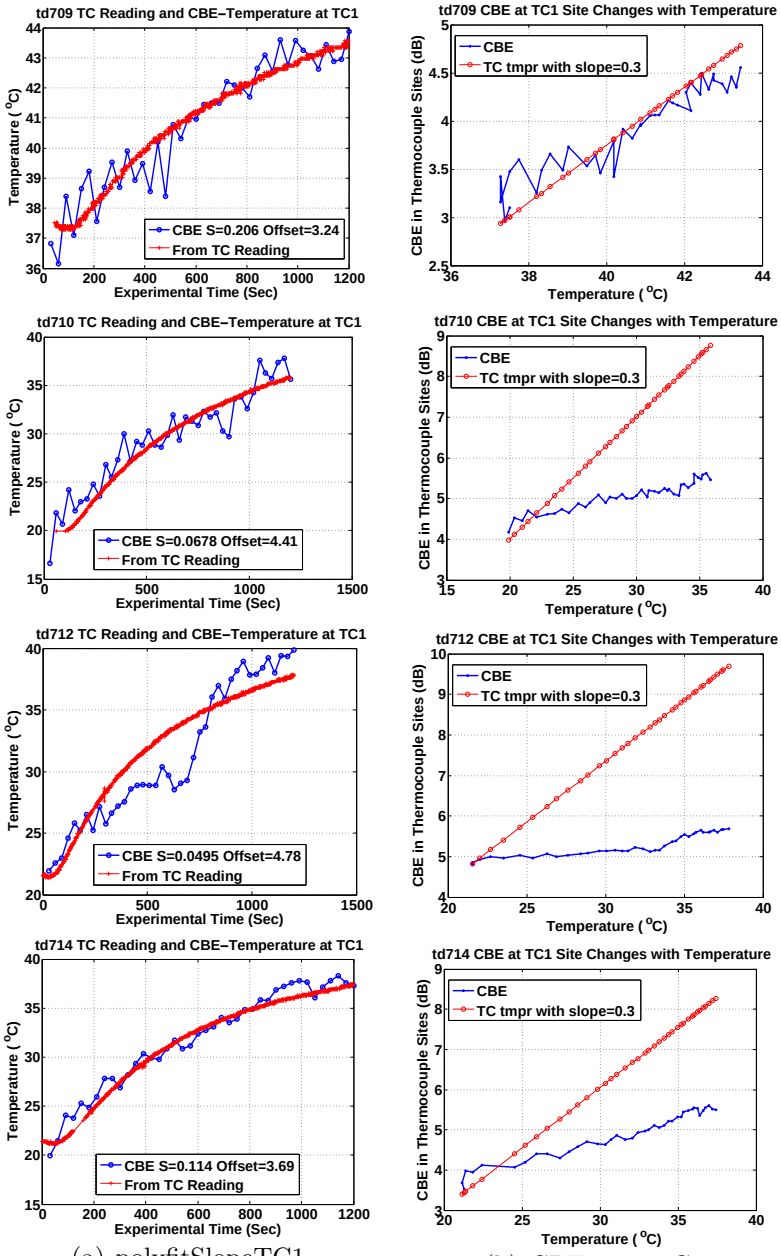
As shown in Fig. 4.40, the average of temperature estimation error at each of the 10 thermocouple locations were plotted. Along with their corresponding standard deviation of the mean. The green line represents the temperature estimation error over the 10 thermocouple sites (10x40 data entries).



(a) td709 TC1 (b) td710 TC8 (c) td711 TC8

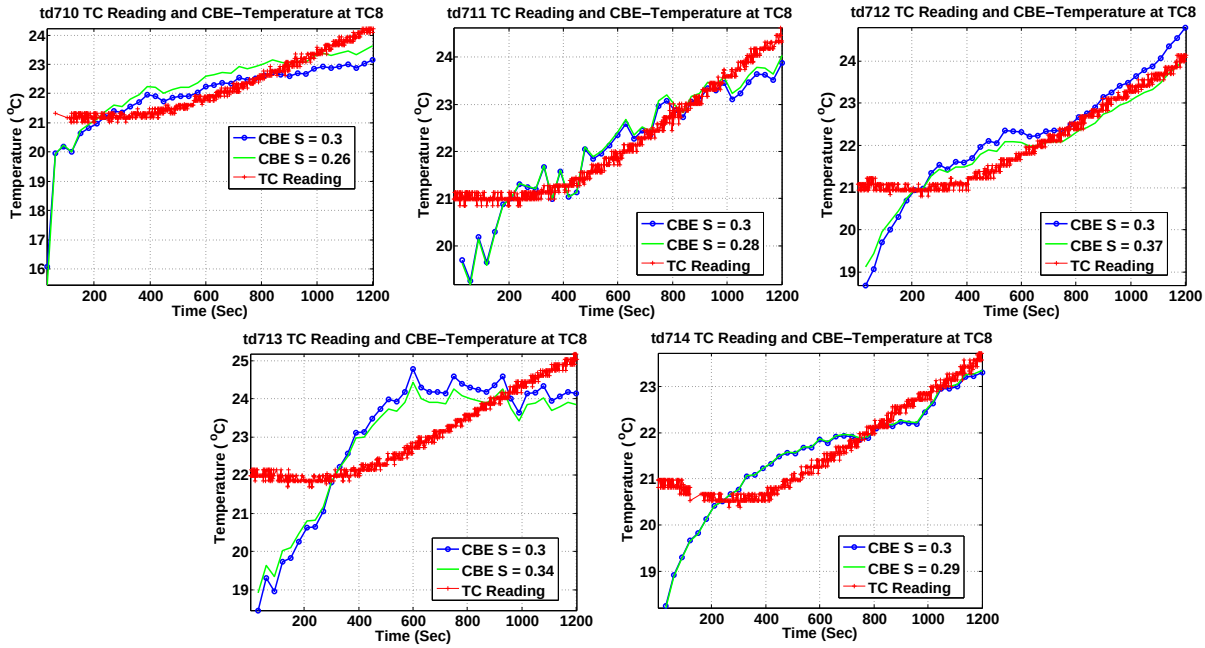
(d) td712 TC8 (e) td713 TC8 (f) td714 TC8

Figure 4.36: Temperature Sensitivity of CBE in each experiment that most closely matched the 0.3 dB/^oC prior calibration value [3]. For each subfigure (a) to (f) Top) The figure on top represents the CBE estimated temperature by first-degree polynomial fit compared to TC reading. The average temperature sensitivity of CBE was 0.29 ± 0.06. Bottom) The figure at bottom demonstrates CBE at TC location as a function of TC reading. Red line with slope = 0.3 dB/^oC serves as a reference for the CBE curve [3].

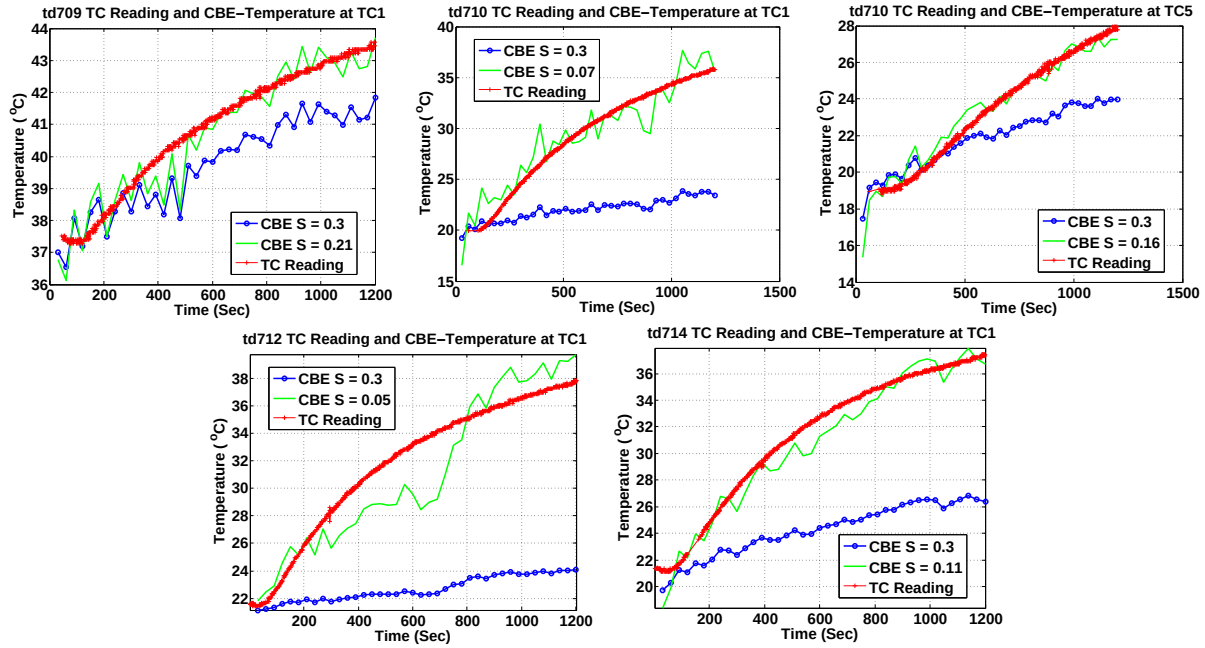


(a) polyfitSlopeTC1 (b) CBETmprTC1

Figure 4.37: CBE-Temperature Sensitivity at 5 TC1 Sites. (a) CBE estimated temperature by first-degree polynomial fit compared to TC1 temperature readings. The average temperature sensitivity of CBE was 0.11 ± 0.07. (b) CBE at TC1 site as a function of TC1 reading. Red line with slope = 0.3 dB/^oC serves as a reference for the CBE curve [3].

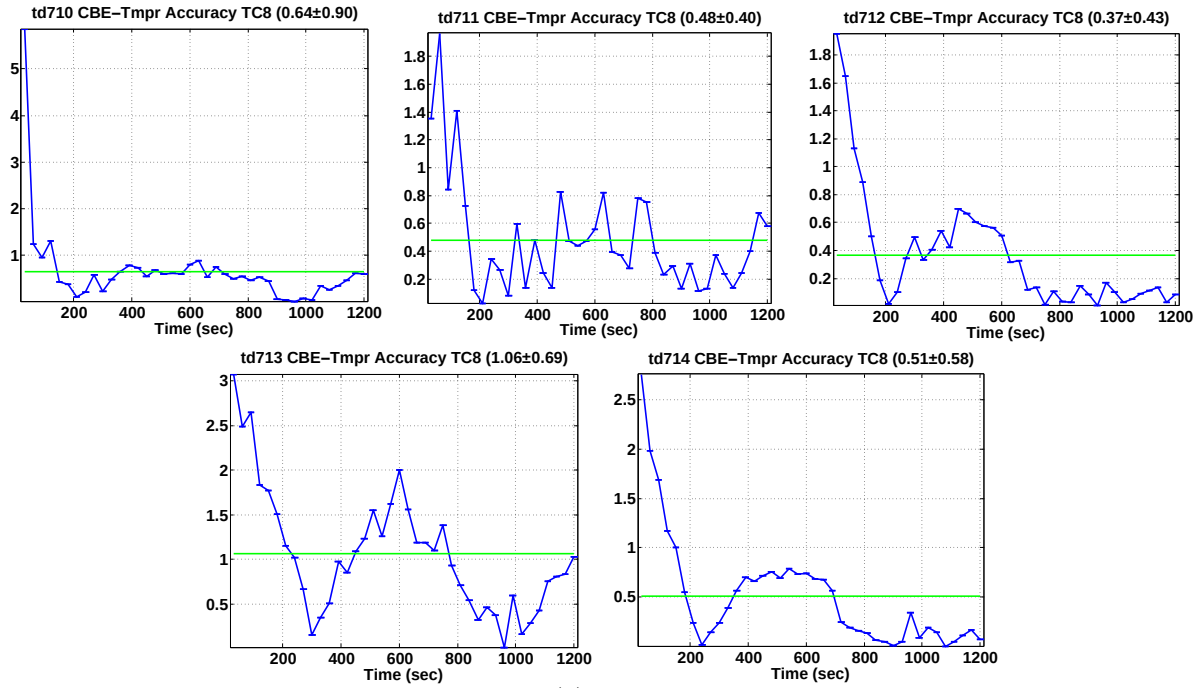


(a) Close to 0.3 dB/^{o}C

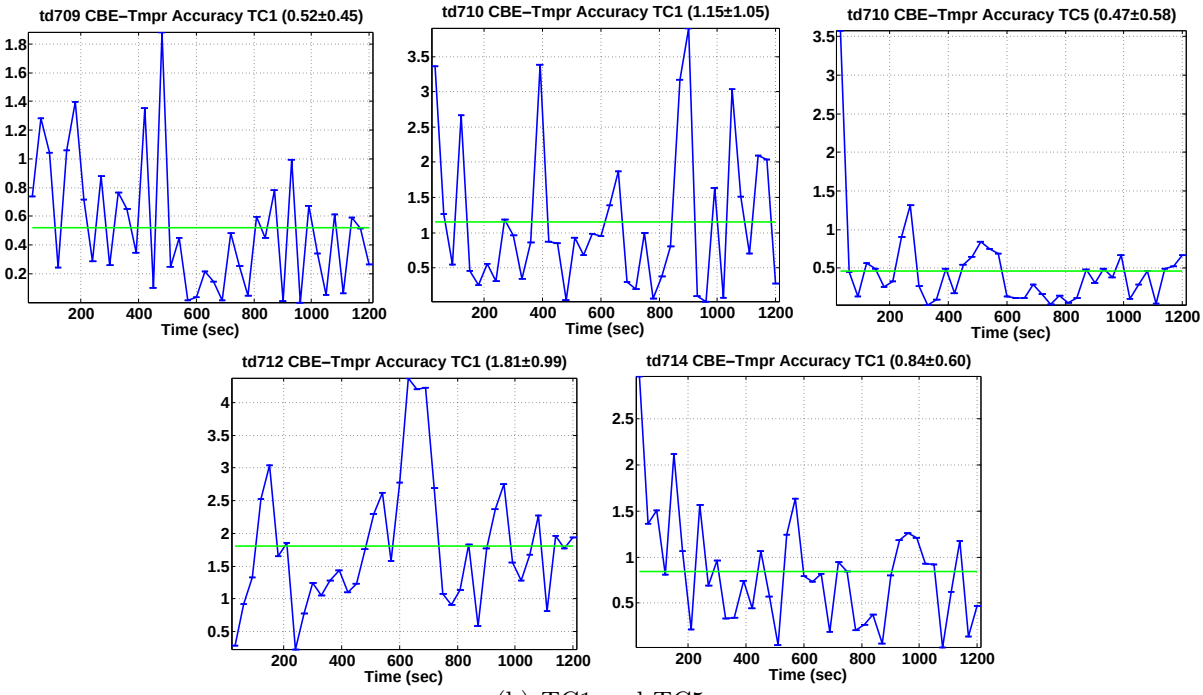


(b) NOT close to 0.3 dB/^{o}C

Figure 4.38: CBE estimated temperature vs thermocouple readings at 10 TC sites. In each figure, the blue curve shows CBE estimated temperature with a CBE-Temperature sensitivity of 0.3 dB/^oC [3] . The green curve represents CBE estimated temperature with CBE-Temperature sensitivity obtained from first-degree polynomial calculation between CBE value and TC reading.



(a) TC8



(b) TC1 and TC5

Figure 4.39: The error of CBE estimated temperature at 10 TC locations as a function of experiment time.

Table 4.1: CBE-Temperature Sensitivity and Estimation Error over 20 min @ 30 s Intervals

Thermocouple	CBE Thermal Sensitivity(dB/^{o}C)	CBE Estimation Error(^{o}C)
td710 TC8	0.26	0.64 ± 0.90
td711 TC8	0.28	0.48 ± 0.40
td712 TC8	0.37	0.37 ± 0.43
td713 TC8	0.34	1.06 ± 0.69
td714 TC8	0.29	0.51 ± 0.58
Average TC8	**0.31±0.045**	**0.61±0.66**
td709 TC1	0.21	0.52 ± 0.45
td710 TC1	0.07	1.15 ± 1.05
td712 TC1	0.05	1.81 ± 0.99
td714 TC1	0.11	0.84 ± 0.60
Average TC1	**0.11±0.071**	**1.08±0.93**
td710 TC5	0.16	0.47 ± 0.58
Total Average	**0.21±0.11**	**0.78±0.81**

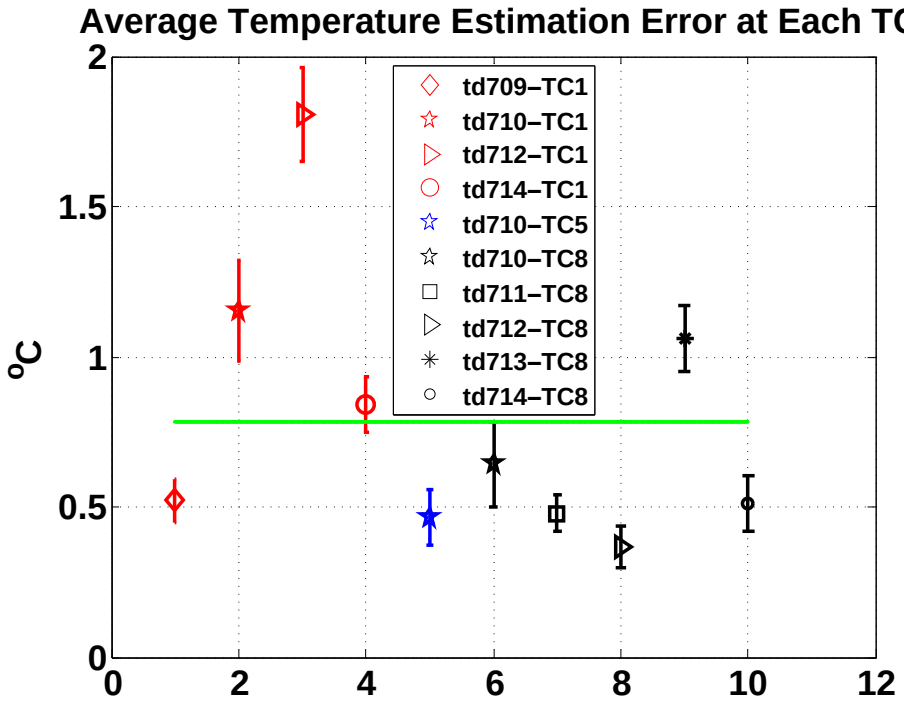


Figure 4.40: Average temperature estimation error at 10 thermocouple sites. Error bar of standard deviation of the mean at each thermocouple was plotted. The green line represents the average estimation error over the 10 thermocouple sites.

Chapter 5

Temperature Imaging in Real Time

In Chapter 4, we demonstrated the 2D ultrasonic temperature imaging system with a potential to yield a comparable quality to the 3D system in CBE-Temperature estimation. Through analyzing each experiment, we saw good correlation coefficient (above 0.99) between adjacent image pairs. With accumulated motions ranging from 0.035mm to 0.5 mm in both axial and lateral direction, the motion between adjacent images was between 4 μm and 12 μm. We also discovered that using the temperature readings of thermocouple (TC) #8 and estimated CBE at TC8 site, the CBE-temperature sensitivity was nearly equal to 0.3 dB/^{o}C with an error less than 1 ^{o}C, which is our objective for temperature imaging (TI) [10].

The analysis results in Chapter 4 illustrated that we can move the ultrasonic temperature imaging (TI) system from 3D to 2D with a simpler experimental setup and faster data acquisition. 2D also means less data to process, which provides the possibility of implementing real-time TI. The key steps in obtaining temperature maps from raw ultrasonic images are motion compensation, CBE calculation, and conversion of CBE to temperature.

Among which, motion compensation (MC) is most time consuming part, and thus the limit to real-time TI. Because MC requires aligning images for a pixel-by-pixel determination of CBE [4], in this chapter, we discuss approaches to reducing the MC time, so that we may be closer to real-time ultrasonic temperature imaging.

5.1 Platforms for Testing Real-Time Imaging

Tab. 5.1 demonstrated the platforms we used to benchmark the "real-time" imaging process. We carried out the motion compensation on each of the platform, to compare the time cost. It's shown in the table that the RTTI platform has the best hardware configuration, and it's also equipped with a GPU processor, which makes RTTI the fastest machine both in CPU and GPU category among the 4 platforms. The Custom Workstation @ Lickenbrock comes after RTTI since both its CPU and GPU are less powerful. Third place is the HP DV6 machine (2-3 times slower than the RTTI CPU), and last is Macbook Pro (about 2 x slower than HP DV6).

Table 5.1: Configuration Comparison among Computers Used for Benchmark Tests

Computer Name	**Processor**	**Configuration**
RTTI	CPU	Intel Core i7-4820K Quad-Core CPU @ 3.70GHz
		RAM: 16.0GB
	GPU	GeForce GTX 770
		Dedicated Video Memory: 2048 MB GDDR5
Custom Workstation @Lickenbrock Technologies, St. Louis Carried out by C.D.Holmes	CPU	Intel Core 2 Quad Q9300, 2.5 GHz processor
		RAM: 8 GB
	GPU	NVidia GTX260
		Dedicated Video Memory: 896 MB
HP DV6	CPU	AMD A8-3520M APU; Radeon HD Graphics
		RAM: 6.0 GB
Macbook Pro	CPU	Intel Core i5, 2.5 GHz
		RAM: 4 GB 1600 MHz DDR3

5.2 Rigid Motion Compensation

We approached to the rigid MC method in first because it involves less computation, and theoretically would cost less time than non-rigid MC. We divided the ROI into several sub-regions as shown in Fig. 5.2. The pixel spacing of the original image is 32 μm axially, and 300 μm laterally. As we discussed earlier this chapter, the motion between adjacent images

ranges from 4 μm to 12μm. To better track the motion, pixel spacing must be interpolated and correlated.

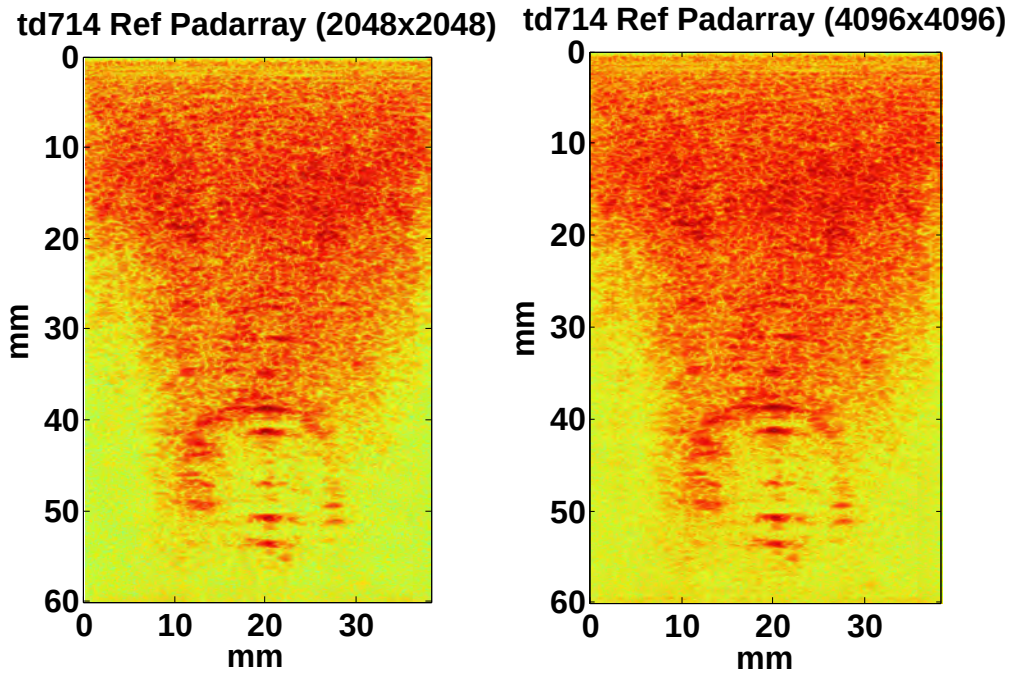


Figure 5.1: Interpolation of the original image (1871x128 pixel) using *padarray*. Left) 2048x2048 pixels. Right) 4096x4096 pixels. Both match the original and appear to be identical by observation.

5.2.1 Interpolation and Correlation Function

Image interpolation was done using *interpft* function from Matlab. While in GPU, *interpft* function is not supported, so we used *padarray* in conjunction with *fft2* and *ifft2* as an alternative method. Tab. 5.2 contains the running time of *padarray+fft2+ifft2* on both CPU and GPU. It shows that the larger the up-sampling scale, the longer it takes the computer to run. When the original image (1871x128) was interpolated to 2048x2048, the pixel spacing was 29 μm axially (1.1x), and 20 μm laterally (16x). Interpolation to 4096x4096 yielded pixel spacing of 13 μm axially (2.4x), and 9 μm laterally (32x) as seen in Fig. 5.1. The smaller the pixel spacing, the better the motion (4-8 μm) can be compensated, however, the

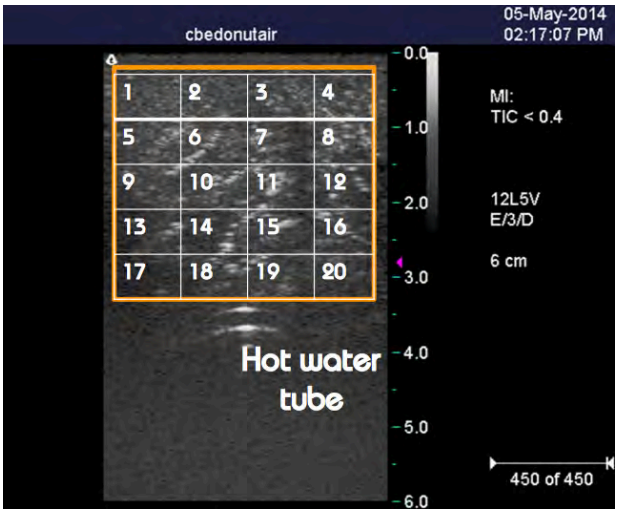


Figure 5.2: 2D rigid motion compensation over 20 subregions (5 x 4 white) over the region of interest (orange). Image pixel spacing was 32 μm in the axial (up / down) direction and 300 μm in the lateral direction (left / right).

longer the processing time. Note that up-sampling an image using *interpft* is slower than *padarray*.

For 2D rigid motion compensation, we divided images into 25 subregions. See Fig. 5.3. Over the ROI we used *normxcorr2* to calculate the correlation coefficient in each subregion. The time cost of running *normxcorr2* over different region size is included in Tab. 5.2 as well. Fig. 5.4 illustrates the running time of *normxcorr2* on both CPU and GPU in bar graphs. We can see that, the larger the region size, the more advantage would be gained through using GPU.

The process of interpolation and correlation took about 12 sec per image on HP DV6. We didn't ran the rigid motion compensation on any GPU platform. As shown in Fig. 3.8 (center), it takes the RTTI CPU 6.9 sec per image to run non-rigid MC, which is faster than the rigid MC ran on HP DV6. It may take 4 sec if the rigid MC was run on RTTI CPU, which makes it 1.75 x faster than non-rigid MC, however, we believe with the implementation of GPU, the speed up from using rigid MC method would be neglectable. Plus, non-rigid MC can render better result. Hence we decided to employ non-rigid MC method in our real-time TI study.

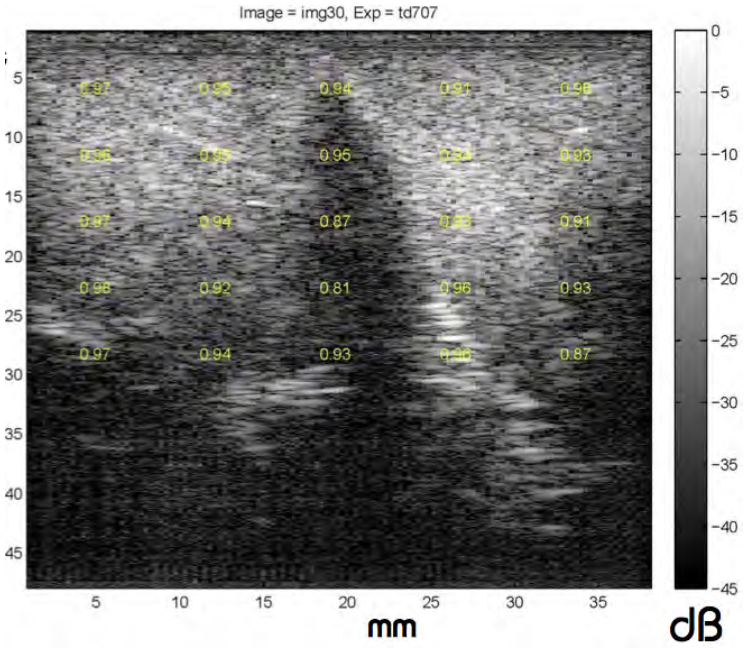


Figure 5.3: The ROI is divided into 25 subregions. The numbers highlighted in yellow represent the value of correlation coefficient at each subregion.

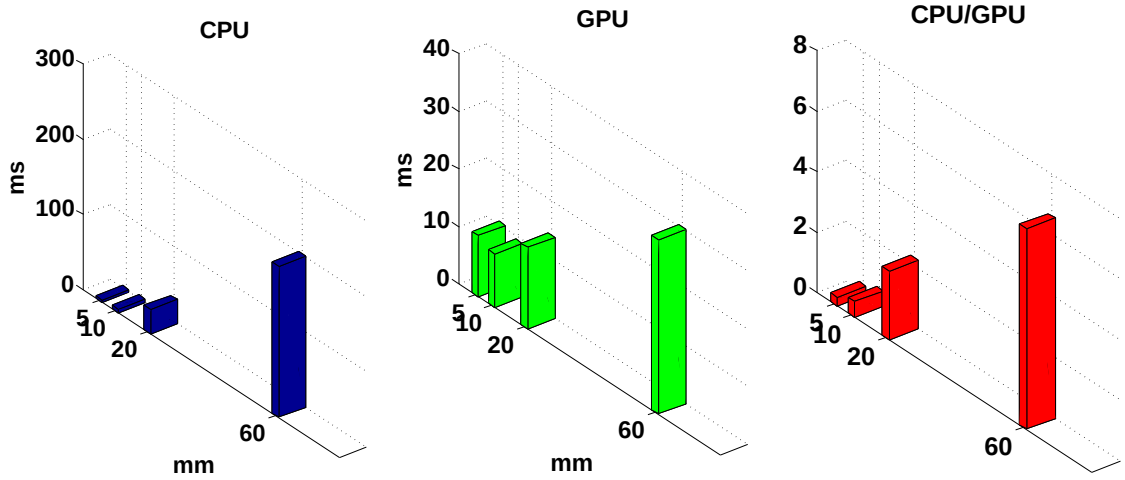


Figure 5.4: Bar graphs of *normxcorr2* runtime on different region size(5x5mm, 10x10mm, 20x20mm, 60x38mm). Left) Runtime on CPU. Center) Runtime on GPU. Right) The ratio of runtime between CPU and GPU.

Table 5.2: Benchmark Runtimes per Iteration for Rigid Motion Compensation Routines

Operation	Region Size	Iterations	CPU	GPU	C/G
Interpolation	60x38 mm		ms	ms	
padarray	1871x128 (px) to 2^{11} x 2^{11}	10	203	6.11	33.2
	1871x128 to 2^{12} x 2^{12}	10	486	75.1	6.47
interpft	1871x128 to 2^{11} x 2^{11}	10	245	N/A	N/A
	1871x128 to 2^{12} x 2^{12}	10	942	N/A	N/A
Motion Detection					
normxcorr2	5x5 (155x16 pxl)	100	3.01	10.6	0.283
	10x10 (310x32)	100	4.71	9.31	0.505
	20x20 (621x67)	100	32.1	14.1	2.27
	60x38 (1871x128)	100	201	30.3	6.63

5.3 Non-rigid Motion Compensation

Since the time cost for non-rigid MC is comparable with rigid MC, whereas, the non-rigid MC could provide a better quality. We decided to move our focus to speed up non-rigid MC to achieve real-time TI.

We are using an algorithm developed by Dr. Trobaugh for non-rigid motion estimation and compensation over region of interest (ROI). The motion field was modeled to vary linearly over ROI and represented as a linear function of the motion at the control points, which were chosen as the 4 corner points of the region [15]. Motion was estimated through searching the displacements at the control points that maximizes the cross correlation between two images. Then interpolate the shift at 4 corner points to ROI to get the motion field of the entire ROI.

The process involves with interpolation (after shift optimization) and correlation cost function. Since the correlation takes very little time, and thus can be eliminated. The major limit becomes the interpolation time during non-rigid MC.

The non-rigid MC routine contains 4 *interp2* functions, two of which are computing the indices of shifted image based on the interpolation of the shift found at 4 corners, we call them *interp2(a)*. Another one we call *interp2(b)*, interpolates to find the values of the original

(before shift) image at indices after shift optimization obtained by *interp2(a)* and return the shifted image. The last *interp2* function was never used, we call it *interp2(c)*. We still investigated it to keep the consistency, since it's always executed during our previous benchmarking. *interp2(c)* returns the shifted image on a refined grid, interleaving 3 interpolates between every element. Tab. 5.4 contains the input and output data size of each *interp2* function.

5.3.1 Interpolation Benchmarks

We carried out a series of test runs to find out the relationships between the three interpolation functions used in non-rigid MC routine and the region size on RTTI CPU and GPU. The results are shown in Tab. 5.3.

Both *interp2(a)* and *interp2(b)* were run 117 times in average per image. This number would vary per experiment, and within the same experiment, it will change with image also. *Interp2(c)* always run once per image. The data in Tab. 5.3 were all from td714.

According to Tab. 5.3, we can see that for each *interp2* function, the larger the region size, the more we can benefit from using GPU. By applying GPU to a 1058x120 pixel region, we can reduce the non-rigid MC time per image to less than 1 second, which is almost real-time.

Table 5.3: Benchmark Runtimes per Iteration for Non-rigid Motion Compensation Routines

Function Name	Region Size(mm)	Iterations	CPU(ms)	GPU(ms)	C/G
interp2(a)	5x5 (155x16 pxl)	1000	0.632	1.41	0.448
ax + lat	10x10 (310x32)	1000	0.972	1.42	0.684
117 times	20x20 (621x67)	1000	2.24	1.82	1.23
	34x36 (1058x120)	**1000**	**5.54**	**1.42**	**3.90**
interp2(b)	155x16	1000	0.232	0.620	0.374
image resample	310x32	1000	0.286	0.638	0.448
117 times	621x67	1000	0.427	0.669	0.638
	1058x120	**1000**	**1.66**	**0.672**	**2.47**
interp2(c)	155x16	100	0.243	0.134	1.81
1 time	310x32	100	0.249	0.139	1.79
	621x67	100	0.246	0.132	1.86
	1058x120	**100**	**0.235**	**0.109**	**2.15**
MotionTool--Nonrigid2D	**1058x120**	**200 (5 x 40)**	**2.16s**	**0.813s**	**2.65**

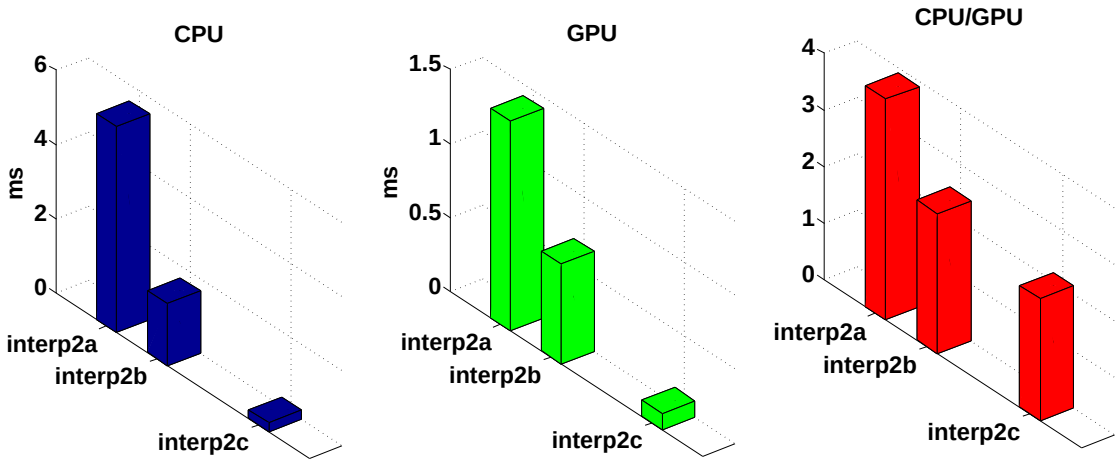


Figure 5.5: Bar Graphs of *interp2* runtime in three different scenarios (*interp2(a)*, *interp2(b)*, *interp2(c)*). See Tab. 5.4 for detail. Left) Runtime on CPU. Center) Runtime on GPU. Right) The ratio of runtime on the CPU versus that on the GPU.

Note that MotionToolnon-rigid2D is the routine that does non-rigid MC, which contains *interp2(a)*, *interp2(b)* and *interp2(c)*. Fig. 5.5 demonstrates the runtime of the three *interp2* functions on CPU and GPU.

5.3.2 Interpolation within Shift Optimization

In the MotionToolNonrigid2D routine, a Matlab© function called *fmincon* is doing the shift optimization. We changed the 'TolFun' threshold, which sets the tolerance criteria that satisfies the optimization algorithm.

The larger the 'TolFun', the less time it takes to run the shift optimization, and hence reduces the non-rigid MC time. In the previous section, 'TolFun' was set to 10^{-5}. When set to 10^{-1} and run on td714 data. The change gave us a 4x speed up on both RTTI CPU and GPU while maintaining an average correlation coefficient above 0.99.

With this speed up, the non-rigid MC time was reduced to 0.22 sec, when run on the RTTI GPU. We believe, with further investigation to the optimization algorithm, we can have more speed up through changing the parameters in the motion compensation optimization routine.

Table 5.4: Input and Output of *Interp2* Functions

Interp2 (linear)	**Input Data Size**	**Output Data Size**
Interpolate shift over the image size Interp2(a)	2x2	1058x120
Resample to match reference image Interp2(b)	1107x125	1058x120
Interpolated values on a refined grid Interp2(c)	1058x120	4229x477

5.4 Summary

In this chapter, we carried out a series of studies aiming to reduce the motion compensation time, which we believe is the main obstacle to achieve real-time temperature imaging.

We focused on ways to reduce the rigid motion compensation time first, because we thought rigid motion compensation (MC) would consume less time than non-rigid method, which

requires much more calculation. Our benchmark studies, however, suggested a different perspective.

We found out that the runtime of non-rigid MC was only 1.7 times slower than rigid MC (4 sec using RTTI CPU platform) per image pair. Consider the far more accurate MC the non-rigid MC produces, and the fact that 1.7x over 4 sec would be negligible with the deployment of GPU, we changed our focus to non-rigid MC.

With further study of how non-rigid MC program works, we found the most time consuming part was the MotionToolNonrigid2D subroutine. This subroutine contains estimating motion at 4 corners of 2D region of interest(ROI), and resample image based on the non-rigid motion found at 4 corners Tab. 5.3 and Tab. 5.4.

Using GPU feature of the Matlab, we managed to reduce the non-rigid MC time to 0.22 sec on RTTI GPU platform. The major barrier to real-time TI no longer lies with motion compensation. The limit now lies with the Terason 3000 system, because Terason 3000 takes about 5 sec to transform an ultrasonic image in its native format to Matlab format before sending it to the TI computer. Therefore, we believe CBE TI can be done at a 1 Hz frame rate with $< 1^{o}$C error if conversion to Matlab in the Terason 3000 can be reduced to less than 0.7 sec.

Chapter 6

Summary and Conclusions

To achieve real-time temperature imaging (TI), a two-computer architecture was implemented in this study, with the Terason 3000 ultrasonic imaging system and a TI machine connected through jtcp internet connection. The TI machine was an HP Envy Phoenix 810 with a GeForce GTX 770 GPU card (RTTI). The TI computer performed motion compensation and extracted temperature images from raw ultrasonic images collected by Terason 3000 system.

Six experiments were carried out on turkey breast specimens during heating with hot water (75 ^{o}C) in 1 cm tube. Total heating time was 1200 sec, with a 30 sec interval between image acquisitions, tissue temperature was monitored at sparse spots with thermocouples. Ultrasonics images were taken in 2D, which means less data acquisition and less computation, compare to 3D CBE TI model carried out by Basu [10].

With the accuracy of CBE TI being 0.8 ± 0.7^oC, we proved the feasibility of using CBE to estimate temperature on 2D images. We managed to reduce the computational time from ultrasonic images to TI while maintaining at least 1^oC accuracy on average.

Benchmark studies were carried out to find the optional combination of motion compensation (MC) quality and the time it costs. We managed to reduce the MC time to less than 0.3 sec per image pair using non-rigid MC method on RTTI GPU.

6.1 Discussion

From the 3D calibration studies done by our group, the temperature sensitivity of CBE was 0.3 dB/^{o}C on average[10]. In this study, we found, over 10 usable sites through 6 experiments, that the average was 0.21 $\pm$ 0.11 dB/^{o}C. We also noticed that the thermocouple sites further away from the heating source, the closer their CBE-temperature sensitivity to 0.3 dB/^{o}C.

The possible causes are: 1) the thermal gradient near the hot-water tube, 2) lack of information in the elevation direction, and 3) the choice of a 6x6 mm^2 region of interest for calculating CBE. Further investigation will be carried out in the future to find out the influence of the above three factors on CBE temperature sensitivity.

The thermocouples need to be re-calibrated or replaced as well. Because the thermocouple readings at different spots of a temperature-equilibrium tissue have an difference of 2 ^{o}C as shown in Fig. 4.7 (td713). Thus, we think the inaccuracy of thermocouples which serve as a calibration source to our temperature estimation using CBE, could be another reason that introduces error the CBE TI.

As shown in Fig. 4.6, there was an initial jump in the CBE value for each experiment. We believe the jump was caused by the inherent noise in the system, whose effect has already been simulated as seen in Fig. 2.2 (Right).

We were also able to reduce the nonrigid MC time to 0.22 sec per image pair using the RTTI GPU. It's also our prediction that the nonrigid motion compensation algorithm can have a further speed up with an improvement in the optimization algorithm itself.

6.2 Conclusions

We reproduced Basu's non-uniform heating with hot-water tube model in 2D [10], obtained similar temperature sensitivity of CBE at sites far from heating source and good estimation accuracy, 0.8$\pm$0.7^{o}C. These results laid the foundation for a faster CBE TI system. Future investigation will be carried out to refine the 2D experiment setup to limit the error introduced by the value of CBE-temperature sensitivity.

Using its CPU, the TI computer updated temperature images using rigid MC in 4 sec, and using more nearly accurate nonrigid MC in 7 sec. Nonrigid MC time was reduced to 0.2 sec using the GPU processor along with optimization of the MC algorithm. Calculation of CBE in MC images and conversion of CBE to TI takes less than an additional 0.1 sec. Therefore, the TI time was reduced to < 0.3 sec.

The limit to real-time CBE TI now lies with the Terason 3000 system. Because it takes about 5 sec to transform an ultrasonic image in its native format to a Matlab file (.mat) before sending it to the TI computer, during which, no other process can be done. As a result, we believe CBE TI can be done at a 1 Hz frame rate with < 1 ^{o}C average error, if conversion to Matlab in the Terason 3000 can be reduced to less than 0.7 sec. This study of 2D CBE TI in real-time demonstrated a very promising future in clinical application for ultrasonic TI.

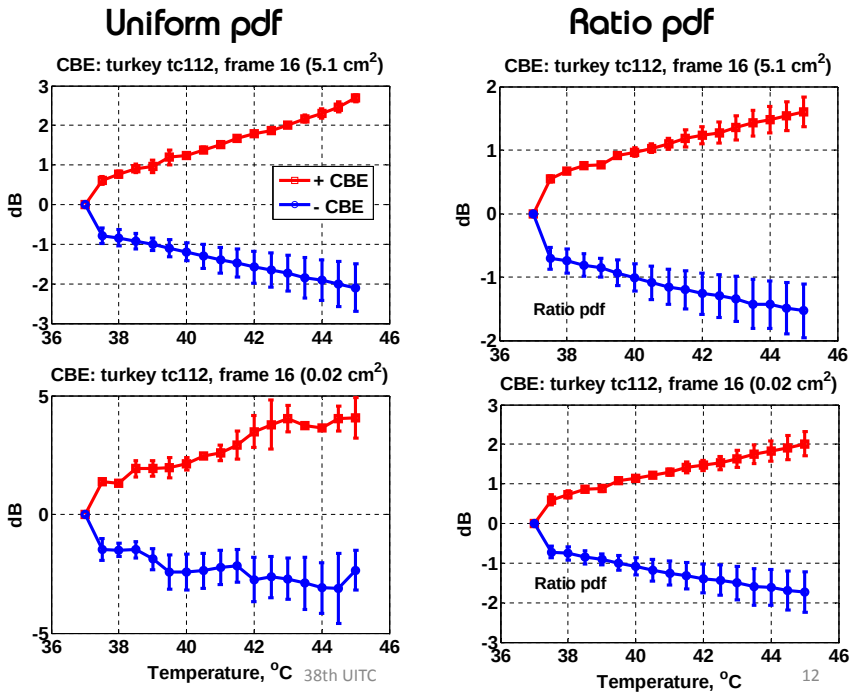


Figure 6.1: Top) For a larger region (51 mm^2), the effect of the ratio pdf is small. Bottom) For a region 25 x smaller (2 mm^2), the ratio pdf effect in linearizing CBE is significant. From Arthur et al. 2013 [8]

6.3 Future Work

In our future investigation, we want to reduce the time cost by the Terason 3000 system to transform a ultrasonic image in its native format (.ult) to Matlab readable format (.mat) to less than 0.7 sec. To achieve this goal, We want to work in collaboration with the Terason Company as partners, to develop a new feature in Terason 3000 system that can do the transformation faster (less than 0.7 sec). Another option is to develop a Matlab routine ourselves that can read .ult files.

In addition, we will implement the ratio probability density function method to calculation CBE [15, 8]. By doing so, we will be able to obtain a more nearly accurate CBE estimation over much small regions of interest than studied here (see Fig. 6.1). This method will allow us to calculate temperature in multiple subregions of the region of interest, which may yield TIs with better spatial resolution without sacrificing temperature accuracy.

Appendix A

Custom Matlab Codes on Terason 3000 System

A.1 Custom Matlab Code for Experiment Controlling

A.1.1 ablwdonut_zwy8nrsvwyz.m

```
% ABLWDONUT_ZWY8WYZ
%   based on THERM3D_SDK_CDH and THERM3D_SDK_ZWY3
%   both from Yuzheng Guo's sdk code
% Temperature Imaging with CBE
% Terason 3000 using software developer's kit for imaging
%   set ablation water temperature, find thermocouple offset & take tc ...
    readings
% Terason linked via tcp to TI machine (Dell 2.4GHz)
%   for 2D motion compensation evaluation via CC -> CBE calculation, TI ...
    display
% Heat source:  Hot water set to temperture in hdr file (>65oC)
% R. Martin Arthur
% 21 July 2013
% 28 April 2014
% Modified by Weiyuan Zhao
% 23 AUG 2014

format compact; clc;
```

```
17
18 % Prepare the experimental setup
19 disp('**EXPERIMENTAL SETUP**');
20 disp('Clamp the vise to the table');
21 disp('Adjust the distance between the thermocouple supporting')
22 disp('  bracket and donut plates to 2.5cm inner surface to inner surface...
       ')
23 disp('Drill a 5 mm hole (cork borer) in the tissue center')
24 disp('Put the tissue between the donut plates')
25 disp('Hand tighten the nylon screws')
26 disp('Use long screw in hot water tube to thread tube through tissue');
27 disp('->Use gel for tissue contact');
28 disp('Put donut assembly in the vise and tighten vise')
29 pause; clc;
30
31 disp('**IMAGING**');
32 disp('Fix the transducer to the motor mount')
33 disp('Put coupling gel on the tissue')
34 disp('Move the transducer to the tissue with the NewPort')
35 disp('IF CONNECTED TO EXPERIMENTAL DATA DIRECTORY, PRESS ANY KEY');
36 pause;
37 disp('Turn on power strip to Dataq and Newport')
38 disp('Turn on the ThermoHaake');
39 disp('Press any key when ready'); pause;
40
41 % Get experiment ID from pwd, make ult image directory, set header file ...
       name
42 currentpath = pwd; expid = currentpath(end-4:end); hfn = [expid '.hdr'];
43
44
45 % Read header
46 %[hdr, par] = rdulthdrabl(hfn);
47 [hdr, par] = rdulthdrabl(hfn);
48 % Parse parameter values from hdr
49 exam = par.exam; exam=fliplr(deblank(fliplr(exam)));
50 expnum = par.expnum;
51 date = par.date;
52 hwtmpr = str2num(par.hwtmpr);
53 % timestep and number of images from header
```

```
54 sss=str2num(par.imgnb); timestep=sss(2); runtime=sss(end); numimgs=...
       runtime/timestep;
55 %timestep=30; numimgs=15; runtime=timestep*numimgs ;
56 %inittmpr = tmprset(1)-0.5;
57 % tctissue = par.tctissue;
58 % tcwater = par.tcwater;
59 tctissue = str2num(par.tctissue);
60 tcwater = str2num(par.tcwater);
61 tcCh = sort([tcwater tctissue]);
62 % thofst = str2num(par.thofst);
63 % eqΔ = str2num(par.eqΔ);
64 % eqtime = str2num(par.eqtime);
65 % elstep = str2num(par.elstep);
66 % elviews = str2num(par.elviews);
67
68 % Make image mat directories and set path to it
69 if exist([currentpath '\ult'])==0; mkdir ult;
70     imgsavepath = [currentpath '\ult']; end;
71 if exist([currentpath '\mat'])==0; mkdir mat;
72     matsavepath = [currentpath '\mat']; end;
73
74 % WinDaq: Open & start Dataq software
75 open_windaq_cdh; windaqhdl = windaqStart;
76
77 % Serial port for Thermo Haake for tc offsets
78 s=serial('COM9','BaudRate',9600,'DataBits',8,'StopBits',1,'Terminator','...
       CR');
79
80 % Make sure all connections are closed at start of experiement
81 disp(' ');
82 disp('HIT RETURN TO CLOSE CONNECTIONS TO ALL PERIPHERAL DEVICES');
83 fclose('all'); disp(' ');
84
85 % Make tcp connection to TI machine (hpdv6)
86 disp('Making connection via tcp to TI machine'); pause;
87 jtcpobj = jtcp('request','172.16.21.167', 60606,'timeout', 20000); % IP ...
       in hdr?
88 disp('Connection made!');
89
90 % Write Exp Id to TI machines
```

```
jtcp('write',jtcpobj, expid)
disp('expid sent');

%send over the header
hh=reshape(hdr,prod(size(hdr)),1);
%jtcp('write',jtcpobj,hfn)
jtcp('write',jtcpobj,hh);
disp('hdr sent');

% Open Terason
disp('Starting Terason'); hTTauto = StartTerasonActx(exam);
%disp('Starting Terason'); hTTauto = StartTerasonActx('cbedonutair');
disp('Terason is now open! Press any key to continue'); pause; clc;
% Path for saving ult and mat image files
imgsavepath = [pwd '\ult']; matsavepath = [pwd '\mat'];

% Go
disp('Press any key when ready');
pause; clc;

% Set Thermo Haake
haakePumpSpeed_cdh(s, 50);
% Circulate water for 3 seconds
disp('Wait 3 seconds for Thermohaake to mix water');
haakeSetTmpr_cdh(s, 10); haakeGo_cdh(s); pause(3); haakeStop_cdh(s)

% Determine the offset for calibrating the thermocouples using the ...
    ThermoHaake internal thermistor
Tmpr_act_wat = haakeGetThermistorTmpr_cdh(s); wTmpr_th = Tmpr_act_wat;

% Set thermocouple offsets, then place thermocouples
init_tc_zwy2; %init_tc_cdh;

% Position thermocouples just outside the Terason scan
% ... to be determined

% Begin experiment by taking reference image
disp('->');
disp('Press any key to take reference image');
pause;
```

```
130
131 % Start timing
132 exptime = clock;
133 figure(1); grid on; hold on;
134
135 % Reads from WINDAQ stream and calculates thermocouple temperatures
136 Tmpr_act = windaqRead_cdh(windaqhdl, tcCh, offset, TCslopes);
137
138 % Take and save reference image
139 timeArchive = []; imageTimes = []; wTmpr_th = []; tcTmpr = [];
140 % Ult and mat ref images
141 filename = [imgsavepath '\0000.ult']; matfilename = [matsavepath '\0000...
       .mat'];
142 % Ult to mat
143 system(['start C:\ultdataacq\acqprog\Ult2Matlab\Ult2Matlab.exe ' ...
       filename ' ' matfilename]);
144 savesglimage_ttauto(hTTauto,filename);
145 % Open Matlab command window
146 commandwindow; pause(2);
147 % Convert ult to mat
148 pause(2)
149 inputemu('key_down', '\WINDOWS')
150 inputemu('key_down', '\SHIFT')
151 inputemu('key_normal', '\TAB')
152 inputemu('key_up', '\SHIFT')
153 inputemu('key_up', '\WINDOWS')
154 inputemu('key_normal', '\ENTER')
155 inputemu('key_normal', '\ENTER')
156 inputemu('key_alt', 'x')
157 pause(1)
158 loadShell = load(matfilename, 'b_data');
159 ref = loadShell.b_data;
160 clear('loadShell')
161
162 % Write ref name
163 jtcp('write', jtcpobj, '0000')
164
165 % Define region for TI
166 disp('Define topleft & bottomright');
```

```
image=ref; %image = interp1(ref', 1:0.096:128)'; % RMA match Terason B-... Mode images
figure(777);
imagesc(20*log10(abs(hilbert(image))));
%axis image;
caxis([30 64])
% Set toplevel with ginput
%[x y] = ginput(1); topLevel = round(y); jtcp('write', jtcpobj, topLevel... ); close(777);
[x y] = ginput(2);
close(777)
rowscols = round([y(1) y(2) x(1) x(2)]);
jtcp('write', jtcpobj, rowscols);

% image size
imsz=size(ref);
jtcp('write',jtcpobj,imsz);

% Write ref
jtcp('write', jtcpobj, ref);
jtcp('write', jtcpobj, runtime);

% Begin experiment -> timestep from hdr, calc numimgs
% timestep=30; numimgs=15; runtime=timestep*numimgs ; % Put these ... timestep and number of images in header
% Start time and image or step number


%startTime=clock; % comment out by wyz on 23/Aug/14
stepNum=1;

disp('Connect hot water tube to the Thermohaake');
pause;

% Set tmpr to hot water value in header
haakeSetTmpr_cdh(s, hwtmpr);
haakeGo_cdh(s);

startTime = clock; % added by wyz on 23/Aug/14
% Main image acquisition loop
```

```
204 while etime(clock, startTime) < runtime
205
206     Tmpr_act_wat = haakeGetThermistorTmpr_cdh(s);
207     plot(etime(clock, startTime),Tmpr_act_wat,'cyan+');
208     wTmpr_th = [wTmpr_th,Tmpr_act_wat];
209
210     Tmpr_act = windaqRead_cdh(windaqhdl, tcCh, offset, TCslopes);
211     plotTmpr_zwy(etime(clock, startTime), Tmpr_act, tctissue, tcwater)
212     tcTmpr = [tcTmpr; Tmpr_act];
213
214     timeArchive = [timeArchive etime(clock, startTime)];
215
216     drawnow;
217
218     if etime(clock, startTime) > timestep * stepNum
219     elapsedtime=etime(clock,startTime)
220
221         filename = [imgsavepath '\' sprintf('%4.4d', timestep*stepNum) '...
                .ult'];
222         matfilename = [matsavepath '\' sprintf('%4.4d', timestep*stepNum...
                ) '.mat'];
223         system(['start C:\ultdataacq\acqprog\Ult2Matlab\Ult2Matlab.exe '...
                 filename ' ' matfilename]);
224         savesglimage_ttauto_zwy(hTTauto,filename);
225         imageTimes = [imageTimes etime(clock, startTime)];
226
227         commandwindow
228         pause(2)
229         inputemu('key_down', '\WINDOWS')
230         inputemu('key_down', '\SHIFT')
231         inputemu('key_normal', '\TAB')
232         inputemu('key_up', '\SHIFT')
233         inputemu('key_up', '\WINDOWS')
234         inputemu('key_normal', '\ENTER')
235         inputemu('key_normal', '\ENTER')
236         inputemu('key_alt', 'x')
237         pause(1)
238         loadShell = load(matfilename, 'b_data');
239         img = loadShell.b_data;
240         clear('loadShell')
```

```
241         % Write img name
242         jtcp('write', jtcpobj, ['img' num2str(timestep * stepNum)]);
243         jtcp('write', jtcpobj, img);
244         %jtcp('write', jtcpobj, elapsedtime);
245         stepNum = stepNum + 1;
246     end
247 eval(['save ' expid 'run']);
248 end
249
250 haakeStop_cdh(s);
251 fclose(s);
252 eval(['save ' expid 'run']);
```

A.1.2 haakeGetThermistorTmpr_cdh.m

```
1  function thermistorTmpr = haakeGetThermistorTmpr_cdh(s)
2  %
3  % HAAKEGETTHERMISTORTMPR_CDH
4  %
5  % Grabs the current temperature of the water that the ThermoHaake
6  % thermistor is reading.
7  %
8
9  %MAR-25-2011 - added try catch block to catch errors when TH read ...
       timesout.
10 % catch makes recursive call to the function to try the read again.
11 try
12     fopen(s);
13     fprintf(s, 'F1');
14     act_val=fscanf(s);
15     if strncmp(act_val, 'F001',1)  == 0
16         thermistorTmpr = str2num(act_val(4:9));
17     else
18         thermistorTmpr = 34.00;
19     end
20 catch
21     warning('There was a error reading from the thermistor')
```

```
22      warning(datestr(now))
23      fclose(s)
24      thermistorTmpr = haakeGetThermistorTmpr_cdh(s);
25  end
26  fclose(s);
```

A.1.3 haakeGo_cdh.m

```
1  function haakeGo_cdh(s)
2  %
3  % HAAKEGO_CDH
4  %
5  % Starts the ThermHaake pump.
6  %
7
8  fopen(s);
9  fprintf(s, 'go');
10 fclose(s);
```

A.1.4 haakePumpSpeed_cdh.m

```
1  function haakePumpSpeed(s, speed)
2  %
3  % HAAKEPUMPSPEED_CDH
4  %
5  % Set Thermohaake pump speed to speed.
6  %
7
8  fopen(s);
9  fprintf(s, ['W PF ' num2str(speed)]);
10 fclose(s);
```

A.1.5 haakeSetTmpr_cdh.m

```
function haakeSetTmpr_cdh(s, tmpr)
%
% HAAKESTOP_CDH
%
% Stops the ThermHaake pump.
%


fopen(s)
fprintf(s, ['W SW ' num2str(tmpr)]); % set temperature
fclose(s)
```

A.1.6 haakeStop_cdh.m

```
function haakeStop_cdh(s)
%
% HAAKESTOP_CDH
%
% Stops the ThermHaake pump.
%

fopen(s);
fprintf(s, 'st');
fclose(s);
```

A.1.7 init_tc_zwy2.m

```
%INIT_TC_ZWY2
%   Set the thermcouple offsets
%   Direct thermocouple placement
```

```
% Zhao Weiyuan
% 8 July 2013

disp('To fix thermocouple temperature offsets, place the thermocouples')
disp('   under the thermistor in the Thermohaake')
disp('Once the thermocouples are in position, press any key')
pause;

load TCSlopes_cdh.mat
%place holder for slopes 9-16
TCslopes = [TCslopes ones(1,20)];
offset = windaqGetOffset_cdh(windaqhdl, tcCh, Tmpr_act_wat, TCslopes);

tmprcheckbefore = windaqRead_cdh(windaqhdl, tcCh, 0.*offset, TCslopes);
tmprcheckafter = windaqRead_cdh(windaqhdl, tcCh, offset, TCslopes);
clc;
disp('TC Check')
disp('    The left column is the TC measurement with no offset ...
    correction.')
disp('    The right column is the TC measurment with the offset ...
    correction.')
tmprcheck = [tmprcheckbefore tmprcheckafter]

disp('Thermocouple offsets are now determined.')
pause; clc;
% TC placement in tissue
disp('Place thermocouples #1 through #7 on one side of the tissue');
%disp('Place thermocouples #8 through #14 on the other side of the ...
    tissue')
pause(1)
disp('Put the thermocouples at depth (¬5mm) in each experimental ...
    position');
disp('    to be sure each thermocouple is visible in the Terason image')...
    ;
disp('Mark thermocouples on screen');
disp('Photograph screen image');
disp('Manually SAVE image of thermocouples as bmp');
disp('MAKE TERASON LIVE');
disp('Pull thermocouples back until just out of image');
disp('Press any key when finished');
```

```
pause;
% disp('');
% disp('Once the thermocouples are in position, press any key')
% pause;

pause(1)
```

A.1.8 open_windaq_cdh.m

```
% OPEN_WINDAQ_CDH
%
%   Open the windaq software

system('"C:\Documents and Settings\All Users.WINDOWS\Start Menu\Programs...
    \WINDAQ\WinDaq Lite Data Acq DI-1000TC.lnk"');

disp('Starting Windaq software.')
disp('Press any key after Windaq software starts completely!');
pause
```

A.1.9 plotTmpr_zwy.m

```
function plotTmpr_zwy(x, tmprs, tctissue, tcwater)
%
% PLOTTMPR
%
% Plot the tmprs from the thermocouples
%

symbols = ['.' '*' '.' '*' '.' '*' '.' '*' '.' '*' '.' '*' '.' '*' '.' '...
    *'];
colors = ['r' 'r' 'g' 'g' 'k' 'k' 'y' 'y' 'b' 'b' 'c' 'c' 'm' 'm' 'r' 'r...
    '];
```

```
10
11 for k = tctissue
12     plot(x, tmprs(k), [colors(floor(k-1)) symbols(mod(k-1,11)+1)]);
13 end
14 for k = tcwater
15     plot(x, tmprs(k), ['b' symbols(mod(k-1,11)+1)]);
16 end
```

A.1.10 rdulthdrabl.m

```
1  % RDULTHDRABL
2  % [hdr, param] = rdulthdrabl(hfn);
3  % Read header file and extract parameters for experiment
4  % R. Martin Arthur
5  % 29 January 2005
6  % 4 December 2010
7  % 15 Jul 2013 - zwy & rma
8
9  function [hdr, param] = rdulthdrabl(hfn)
10
11 % Read header
12 fid = fopen(hfn, 'r');
13 hdr = hfn;
14 while feof(fid) == 0
15     hdr = str2mat(hdr, fgetl(fid));
16 end
17 fclose(fid);
18 param = struct;
19
20 % Parse hdr
21 ln=3; expnum=deblank(hdr(ln,end-6:end)); param.expnum=expnum;
22 ln=ln+1; date=deblank(hdr(ln,end-10:end)); param.date=date;
23 % Imaging system
24 ln=ln+2; exam=deblank(hdr(ln,end-12:end)); param.exam=exam;
25 ln=ln+4; sr=deblank(hdr(ln,end-3:end)); param.srate=sr;
26 ln=ln+1; cf=deblank(hdr(ln,end-4:end)); param.cfreq=cf;
27 ln=ln+1; sos=deblank(hdr(ln,end-5:end)); param.sos=sos;
```

```
ln=ln+1; pers=deblank(hdr(ln,end-2:end)); param.persist=pers;
% Experiment

% ln=ln+2; tmprset=deblank(hdr(ln,end-10:end)); param.tmprset=tmprset;
% ln=ln+1; tcsfile=deblank(hdr(ln,end-10:end)); param.tcsfile=tcsfile;
ln=ln+2; hwtmpr=deblank(hdr(ln,end-5:end)); param.hwtmpr=hwtmpr;
ln=ln+1; imgnb=deblank(hdr(ln,end-9:end)); param.imgnb=imgnb;
ln=ln+1; tccal=deblank(hdr(ln,end-11:end)); param.tccal=tccal;
ln=ln+1; tcwater=deblank(hdr(ln,end-4:end)); param.tcwater=tcwater;
ln=ln+1; tctissue=deblank(hdr(ln,end-5:end)); param.tctissue=tctissue;
% ln=ln+1; thofst=deblank(hdr(ln,end-4:end)); param.thofst=thofst;
% ln=ln+1; eqΔ=deblank(hdr(ln,end-4:end)); param.eqΔ=eqΔ;
% ln=ln+1; eqtime=deblank(hdr(ln,end-3:end)); param.eqtime=eqtime;
% ln=ln+1; elstep=deblank(hdr(ln,end-4:end)); param.elstep=elstep;
% ln=ln+1; elviews=deblank(hdr(ln,end-3:end)); param.elviews=elviews;
```

A.1.11 windaqRead_cdh.m

```
function tmprs = windaqRead_cdh(windaqHdl,channels,offset, slope)
%
% WINDAQREAD_CDH
%
% This function reads data (temperature) from WINDAQ stream and ...
   calculates
% the temperatures of the Thermocouples based on a linear equation:
%
%                            y = mx + b
%
% Where y is our observed value, b is our offset, and m is our slope.
% Therefore, x is our corrected value.
%
% You can also think of it as a system:
%                           ---
%                   x -> | H | -> y       H: y = mx+b
%
% windaqHdl: handle of WinDaq (Active X) control.
% channels: index of channel from which the data is read\
```

```
% refTmpr: should be the tmpr of the water. This is our initial ...
    correction.
% TcErrSlop: slopes for thermocouples.


numChannels = length(channels);

% Grab 25 tmpr readings in a matrix, then average the readings to a ...
    vector
% of the averages.
for k = 1:25
    for i = 1:numChannels
        %Dataq indexes starting at 0, hence, to get TC 1, pass in 0
        temp(i,k) = GetScaledData(windaqHdl, channels(i)-1);
    end
end
tmpmean = mean(temp, 2);


% x = (y - b)/m
tmprs = (tmpmean - offset)./slope(channels)';
```

A.1.12 windaqStart.m

```
% Function name: StartTerasonActx
% Descript: This function starts Terason 3000 in Matlab as external
% command. Then, create an ActiveX control for TTAutomation associated ...
    with
% the running Terason application. Desired exam will be loaded using ...
    this
% control.
% Input: name of exam. e.g. 'cbe'.
% Output: handle of the TTAutomate control.
% Author: Yuzheng Guo
% Date: 1/15/2009

function hTTauto = StartTerasonActx(exam)
```

```
12
13 if (nargin≠1)
14     disp('There should be one input');
15     exit;
16 end
17
18 if (¬isstr(exam))
19     disp('Input should be a sting of exam name');
20     exit;
21 end
22
23 % start Terason
24 %! C:\Program Files\Teratech\Terason 3000\Ultrasound.exe &
25 system('C:\Program Files\Teratech\Terason 3000\Ultrasound.exe &');
26 pause(15);
27
28 % create TTAutomate control
29 hTTauto = actxcontrol('TTAUTOMATE.TTAutomateCtrl.1');
30 if (¬OpenUltrasound(hTTauto))
31     disp('Terason can not be opened!');
32     exit;
33 end
34
35 % load exam
36 if (¬LoadPreset(hTTauto,exam))
37     disp('Exam can not be loaded!');
38     exit;
39 end
```

A.2 Matlab Control Functions using the Terason Software Developer's Kit (SDK)

A.2.1 StartTerasonActx.m

```
% Function name: StartTerasonActx
% Descript: This function starts Terason 3000 in Matlab as external
% command. Then, create an ActiveX control for TTAutomation associated ...
    with
% the running Terason application. Desired exam will be loaded using ...
    this
% control.
% Input: name of exam. e.g. 'cbe'.
% Output: handle of the TTAutomate control.
% Author: Yuzheng Guo
% Date: 1/15/2009

function hTTauto = StartTerasonActx(exam)

if (nargin≠1)
    disp('There should be one input');
    exit;
end

if (¬isstr(exam))
    disp('Input should be a sting of exam name');
    exit;
end

% start Terason
%! C:\Program Files\Teratech\Terason 3000\Ultrasound.exe &
system('C:\Program Files\Teratech\Terason 3000\Ultrasound.exe &');
pause(15);

% create TTAutomate control
hTTauto = actxcontrol('TTAUTOMATE.TTAutomateCtrl.1');
if (¬OpenUltrasound(hTTauto))
    disp('Terason can not be opened!');
    exit;
end

% load exam
if (¬LoadPreset(hTTauto,exam))
    disp('Exam can not be loaded!');
    exit;
```

```
39 end
```

A.2.2 savesglimage_ttauto_zwy.m

```
1  % Function name: savesglimage_ttauto
2  % Descript: Save a single 2D image using Terason SDK.
3  % Input: handle of the TTAutomate control. image filename.
4  % Author: Yuzheng Guo
5  % Date: 1/15/2009
6
7  function savesglimage_ttauto(hTTauto,filename)
8
9  % Average images
10 %SetPersistence(hTTauto,0);
11 prs=GetPersistence(hTTauto); prs
12
13 % freeze image
14 if(¬FreezeImage(hTTauto))
15     disp('Can not freeze image!');
16     return;
17 end
18
19 % save image
20 if(¬SaveUltrasoundFile(hTTauto,filename,0))
21     disp('Can not save file!');
22     return;
23 end
24
25 % Resume live image
26 if(¬ResumeLiveImaging(hTTauto))
27     disp('Can not resume live image!');
28     return;
29 end
```

A.2.3 saveloop_ttauto.m

```
% Function name: savesglimage_ttauto
% Descript: Save a single 2D image using Terason SDK.
% Input: handle of the TTAutomate control. image filename.
% Author: Yuzheng Guo
% Date: 1/15/2009

function savesglimage_ttauto(hTTauto,filename)

% Average images
%SetPersistence(hTTauto,0);
prs=GetPersistence(hTTauto); prs

% freeze image
if(¬FreezeImage(hTTauto))
    disp('Can not freeze image!');
    return;
end

% save image
if(¬SaveUltrasoundFile(hTTauto,filename,0))
    disp('Can not save file!');
    return;
end

% Resume live image
if(¬ResumeLiveImaging(hTTauto))
    disp('Can not resume live image!');
    return;
end
```

Appendix B

Temperature Imaging Code on GPU based Computer

B.1 rttiwyz2.m

```
% RTTIWYZ
% Real-time CBETI
% Run from anywhere.  Connects to c:\ultthrm\expid
% R. Martin Arthur
% 14 Aug 14
% Weiyuan Zhao
% 23 Aug 14
format compact

%% Make jtcp connection with terason
disp('making connection')
jtcpobj = jtcp('accept', 60606, 'timeout', 20000);
disp('Connection made')

%% Obtain exp id
disp('Waiting for expid')
while true
    msg = jtcp('read', jtcpobj);
    if ischar(msg)
        expid = msg;
        disp('Experiment ID received')
```

```
22          break
23      end
24  end
25
26  %% Obtain header file
27  disp('Waiting for header file')
28  while true
29      msg = jtcp('read', jtcpobj);
30      if ischar(msg)
31          hdr = msg;
32          disp('Header file received')
33          break
34      end
35  end
36
37  %% Make and connect to experiment directory
38  eval(['cd c:\ultthrm\']); eval(['mkdir ' expid]); eval(['cd ' expid]);
39
40
41  %% Obtain ref name -> [0000.mat]
42  disp('Waiting for ref name')
43  while true
44      msg = jtcp('read', jtcpobj);
45      if ischar(msg)
46          image_name = msg;
47          disp('Ref name received')
48          break
49      end
50  end
51
52  %% Get row & scols for the region to process
53  disp('Waiting for region coordinates');
54  while true
55      msg = jtcp('read', jtcpobj);
56      if all([¬isempty(msg) (length(msg) == 4)])
57          rowscols = msg;
58          disp('Rows / cols of analysis region set');
59          break;
60      end
61  end
```

```
62
63 %% Obtain image size
64 disp('Waiting for image size [rows cols]')
65 while true
66     msg = jtcp('read', jtcpobj);
67     if all([¬isempty(msg) (length(msg) == 2)])
68         image_size = msg;
69         disp('Image size received')
70         break
71     end
72 end
73
74 %% Start real-time image acquision
75 disp('Start realtime acquision beginning with the reference image')
76 k=1; % Index for ref image
77 figure;
78
79 %% Obtain ref image -> 0000,mat
80 disp('Waiting for ref image [0000.mat].')
81 while true
82     ref = jtcp('read', jtcpobj);
83     if all([¬isempty(ref) image_size'==size(ref)])
84         disp('Reference Image Received')
85         break
86     end
87 end
88
89 img(k,:,:)=ref; k=k+1;
90
91 %% Acquire runtime
92 disp('Wating for runtime')
93 while true
94     runtime = jtcp('read',jtcpobj);
95     if(¬isempty(runtime))
96         disp('runtime received')
97         break
98     end
99 end
100
101 %% Acquire & process (MC, motion, CBE, TI) next image
```

```
lookingForImagename = true; elapsedtime = 0;
while elapsedtime < (runtime-1);
    msg = jtcp('read', jtcpobj);
    if lookingForImagename
        if ischar(msg)
            image_name = msg;
            lookingForImagename = false;
        end
    else % Update img array & compensate for motion
        if all([¬isempty(msg) image_size'==size(msg)])
            t0 = clock;
            img(k,:,:)=msg; k=k+1;
            im2=squeeze(img(k-1,:,:)); im1=squeeze(img(k-2,:,:));
            [im2mc,shall]=Nonrigid2dip(im2,im1,rowscols,'cubic');
            subplot(121); imagesc(20*log10(abs(hilbert(im2)))); colorbar...
                ;
            subplot(122); imagesc(20*log10(abs(hilbert(squeeze(im2mc))))...
                );
            colorbar;
            title(image_name); drawnow;
            disp(['Processing time = ' num2str(etime(clock, t0))]);
            lookingForImagename = true;
            disp(['Done with ' image_name]);
            timeString = image_name(4:end);
            elapsedtime = str2num(timeString);
        end
    end
end; % New image loop

%% Save tirun file
clear jtcpobj;
eval(['save ' expid 'tirun']);
disp([expid ' Complete!']);
```

B.2 Nonrigid2dip.m

```
function [im2mc,shall]=Nonrigid2dip(im1,im2,rowscols,interp)
% Nonrigid2dpi(im1,im2,'cubic')
% Compare im2 to im1 which are full Terason images after selecting row /...
    col region
% interp: choose 'cubic' or 'spline' interpolation
% im2mc is shifted im2, shall is axial and lateral shift
% Yuzheng Guo
% Modified from NonrigidAll created by Dr. Trobaugh
% R. Martin Arthur
% 18 August 2014
% tic;[im2mc,shall]=Nonrigid2dip(img2,img1,'cubic');toc

motref='p';
roff=25; %roff = input('Enter axial offset: ');
coff=3; %coff = input('Enter lateral offset: ');

im1abs=abs(hilbert(im1)); im2abs=abs(hilbert(im2));

%----- Estimate the motion based first shifted images
c = normxcorr2(im2abs,im1abs);
[maxi maxj] = find(c==max(c(:)));
shift0(1,:,:) = (size(c,2)+1)/2 - maxj; shift0(2,:,:) = (size(c,1)+1)/2 ...
    - maxi;
shift0 = zeros(2,2,2);

if strcmp(motref,'r')
        if strcmp(interp,'cubic')
            shall(i,:,:,:) = MotionToolNonRigid2d(thrfloop(rows,cols...
                ,1,1), thrfloop(rows,cols,1,i+1),roff,coff,shift0);
        elseif strcmp(interp,'spline')
            shall(i,:,:,:) = MotionToolNonRigid2d_spline( thrfloop(rows,...
                cols,1,1),thrfloop(rows,cols,1,i+1),roff,coff,shift0);
        else
            disp('Interpolation method incorrect.');
            disp('Please use cubic or spline interpolation.');
        end
  else
        if strcmp(interp,'cubic')
            shall = MotionToolNonRigid2d(im2,im1,roff,coff,shift0);
        elseif strcmp(interp,'spline')
```

```
37              shall(i,:,:,:) = MotionToolNonRigid2d_spline( im2,im1,roff,...
                    coff,shift0);
38          else
39              disp('Interpolation method incorrect.');
40              disp('Please use cubic or spline interpolation.');
41          end
42  end
43
44
45  %------- Motion compensation on envelope
46  rc=rowscols;
47  ROI = [coff rc(end)-rc(end-1)-coff+1; roff rc(2)-rc(1)-roff+1];
48  [indx,indy] = meshgrid(ROI(1,:),ROI(2,:));
49  y2(1,:,:) = Transform2DIM(im2,0*squeeze(shall),indx,indy);
50  y2abs(1,:,:) = abs(hilbert(y2(1,:,:)));
51  im2mc=y2;
52  end
```

B.3 MotionToolNonRigid2d.m

```
1  function shift = MotionToolNonRigid2d(im1,im2,roff,coff,shift0)
2  % Modified from MotionToolNonRigid created by Dr. Trobaugh
3  % This function uses cubic interpolation
4  % Yuzheng Guo
5  % Input: im1 -- referene image for motion estimation
6  %        im2 -- shifted image
7  %        roff -- axial offset for motion compensation
8  %        coff -- lateral offset for motion compensation
9  %        shift0 -- initial shift
10
11 % use interp2 to find displacement field over region of interest from ...
       samples
12
13 % also uses interp2 to interpolate function over region of interest?
14
15
```

```
% set region
% r1 = 1455; r2 = 1480; c1 = 81; c2 = 95; % for te145
% r1=1312; r2=1562; c1=26; c2=59; % for te159 from rma
% r1 = 1412; r2 = 1412+127; c1 = 41; c2 = 59; % updated for te159
mapmax = 1000;

% define bounds for search (searched image bounds and optimization ...
    bounds)
% roff = 12; coff = 12;
ROI = [coff     size(im1,2)-coff;
        roff    size(im1,1)-roff];

% numimages = size(thrfloop,4);

TypicalShift = zeros(2,2,2);
TypicalShift(:,:,2) = .5*ones(2,2);

% set optimization definitions
options = optimset('DiffMinChange',.1,'DiffMaxChange',.5, ...
    'LargeScale','off','Display','iter','FunValCheck','on','TolFun',1e...
        -5, ... % );
    'TypicalX',TypicalShift); % add typical values

% % % figure(2), clf
% % % im1interp = interp2(im1,2); im2interp = interp2(im2,2);
% % % subplot(121), imagesc(im1interp,[-mapmax mapmax]);
% % % title('Image 1 - Original')
% % % subplot(122), imagesc(im2interp,[-mapmax mapmax]);
% % % title('Image 2 - Original')

r1 = roff; r2 = size(im1,1)-roff; c1 = coff; c2 = size(im1,2)-coff;
im1cmp = im1(r1:r2,c1:c2); im2origcmp = im2(r1:r2,c1:c2);
% im1cmpinterp = interp2(im1cmp,2); im2origcmpinterp = interp2(...
    im2origcmp,2);
% % % figure(3), clf
% % % subplot(121); imagesc(im1cmpinterp-im2origcmpinterp,[-mapmax ...
    mapmax]);
% % % title('Difference Im1-Im2 (over ROI)')

drawnow
```

```
52
53 % shift0 = zeros(2,2,2); % initial estimate of shift assumes values at ...
       four corners of ROI to start
54 % shift0(2,:,:) = 0.5*ones(2,2);
55 % shift0(1,:,:) = -3*ones(2,2);
56 % shift0(1,:,:) = [0.0435,-0.0140; 0.0924,-0.1578];
57 % shift0(2,:,:) = [-0.0188,-0.2382;0.0757,-0.1637];
58 [indx,indy] = meshgrid(ROI(1,:),ROI(2,:));
59
60 % unconstrained version
61 % shift = fminunc(@(shift) CostFunction(reshape(shift,2,2,2),indx,indy,...
       im1,im2),shift0,options);
62
63 % constrained version for constraints on shift
64 A = []; B = []; % linear constraints
65 Aeq = []; Beq = []; % equality constraints
66
67 % bounds
68 % maxshift = 3;
69 % LB = -maxshift*ones(size(shift0)); % lower bound on shift
70 % UB = maxshift*ones(size(shift0)); % upper bound on shift
71 LB(1,:,:) = -(coff-1)*ones(2,2); % lower bound on shift
72 LB(2,:,:) = -(roff-1)*ones(2,2); % lower bound on shift
73 UB(1,:,:) = (coff-1)*ones(2,2); % lower bound on shift
74 UB(2,:,:) = (roff-1)*ones(2,2); % lower bound on shift
75 NONLCON = []; % nonlinear constraints
76 shift = fmincon(@(shift) CostFunction(reshape(shift,2,2,2),indx,indy,im1...
       ,im2),shift0,A,B,Aeq,Beq,LB,UB,NONLCON,options);
77 shift = reshape(shift,2,2,2);
78
79
80 % % % figure(1),clf
81 % % % quiver(interp2(squeeze(-shift(1,:,:)),2),interp2(squeeze(-shift...
       (2,:,:)),2),2);
82 % % % title('Displacement estimate')
83
84 im2shift = Transform2D(im2,shift,indx,indy);
85 im2shint = interp2(im2shift,2);
86
```

```
87 % % % figure(3), subplot(122), imagesc(im1cmpinterp-im2shint,[-mapmax ...
       mapmax]);
88 % % % title('Difference Im1-Im2shift')
89
90 drawnow
91 return
92
93
94 function nxc = CostFunction(shift,shiftindx,shiftindy,y1,y2)
95 % compute normalized cross-correlation of y1 and shifted y2
96 % t1=clock;
97 y2sh = Transform2D(y2,shift,shiftindx,shiftindy);
98 xreg = shiftindx(1):shiftindx(end); % THIS SECTION likely to cause BUGS ...
       laterMot
99 yreg = shiftindy(1):shiftindy(end);
100 y1cmp = y1(yreg,xreg); % use comparable region of y1 for compare
101
102 nxc = sum(sum(y1cmp.*y2sh))/sqrt(sum(sum(y1cmp.^2))*sum(sum(y2sh.^2)));
103
104 nxc = -nxc; % make negative to minimize
105 % t2=clock;
106 % disp(['cost fun time: ' num2str(etime(t2,t1))]);
107 return
108
109 function yshift = Transform2D(y,shift,shiftindx,shiftindy)
110 % transform y by shift where shift is sampled at locations in shiftindx,
111 % shiftindy. xreg and yreg define region of interest in shifted y
112
113 [xloc,yloc] = meshgrid([1:size(y,2)],[1:size(y,1)]); % coordinates of ...
       original y
114
115 [xreg,yreg] = meshgrid([shiftindx(1,1):shiftindx(2,2)],[shiftindy(1,1):...
       shiftindy(2,2)]);
116
117 % compute indices of shifted y based on interpolation of samples of ...
       shift
118 indx = xreg-interp2(shiftindx,shiftindy,squeeze(shift(1,:,:)),xreg,yreg,...
       'linear');
119 indy = yreg-interp2(shiftindx,shiftindy,squeeze(shift(2,:,:)),xreg,yreg,...
       'linear');
```

```
120
121 yshift = interp2(xloc,yloc,y,indx,indy,'cubic');
122
123 return
```

B.4 Transform2DIM.m

```
1
2 function yshift = Transform2DIM(y,shift,shiftindx,shiftindy)
3 % transform y by shift where shift is sampled at locations in shiftindx,
4 % shiftindy. xreg and yreg define region of interest in shifted y
5 % This is created by Dr. Trobaugh
6
7 [xloc,yloc] = meshgrid([1:size(y,2)],[1:size(y,1)]); % coordinates of ...
     original y
8
9 [xreg,yreg] = meshgrid([shiftindx(1,1):shiftindx(2,2)],[shiftindy(1,1):...
     shiftindy(2,2)]);
10
11 % compute indices of shifted y based on interpolation of samples of ...
     shift
12 indx = xreg-interp2(shiftindx,shiftindy,squeeze(shift(1,:,:)),xreg,yreg,...
     'linear');
13 indy = yreg-interp2(shiftindx,shiftindy,squeeze(shift(2,:,:)),xreg,yreg,...
     'linear');
14
15 yshift = interp2(xloc,yloc,y,indx,indy,'cubic');
16
17 return
```

References

[1] A Anand, D Savery, and C Hall. Three-dimensional spatial and temporal temperature imaging in gel phantoms using backscattered ultrasound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 54(1):23–31, January 2007.

[2] RM Arthur. Temperature imaging using ultrasound. In EG Moros, editor, *Physics of Thermal Therapy: Fundamentals and Clinical Applications*, Imaging In Medical Diagnosis And Therapy, chapter 13, pages 1–15. Taylor & Francis, 2012.

[3] RM Arthur, D Basu, Y Guo, JW Trobaugh, and EG Moros. 3d in vitro estimation of temperature using the change in backscattered ultrasonic energy. *IEEE Transactions on UFFC*, 57(8):1724–1733, Aug 2010. PMID: 20679004.

[4] RM Arthur, CD Holmes, and W Zhao. Real-time ultrasonic thermometry based on the change in backscattered energy (cbe). In *31st Society for Thermal Medicine Meeting, Minneapolis, MN, 6-10 May 2014*, May 2014.

[5] RM Arthur, WL Straube, JD Starman, and EG Moros. Noninvasive temperature estimation based on the energy of backscattered ultrasound. *Medical Physics*, 30(6):1021–1029, June 2003.

[6] RM Arthur, WL Straube, JW Trobaugh, and EG Moros. Non-invasive estimation of hyperthermia temperatures with ultrasound. *International Journal of Hyperthermia*, 21(6):589–600, September 2005.

[7] RM Arthur, WL Straube, JW Trobaugh, and EG Moros. In vivo change in ultrasonic backscattered energy with temperature in motion-compensated images. *International Journal of Hyperthermia*, 24(5):389–398, August 2008.

[8] RM Arthur, JW Trobaugh, Y Guo, and B Zhao. Signal-to-noise improvement in ultrasonic thermometry based on the change in backscattered energy. In *Ultrasonic Imaging*, volume 35, page 247. 38th International Symposium on Ultrasonic Imaging and Tissue Characterization, 2013.

[9] RM Arthur, JW Trobaugh, WL Straube, and EG Moros. Temperature dependence of ultrasonic backscattered energy in motion-compensated images. *IEEE Transactions On Ultrasonics, Ferroelectrics, And Frequency Control*, 52(10), October 2005.

[10] Debomita Basu. *3D Temperature Imaging Using Ultrasonic Backscatter Energy during Non-uniform Tissue Heating*. PhD thesis, Washington University in Saint Louis, May 2010.

[11] MJ Borrelli, LL Thompson, CA Cain, and WC Dewey. Time-temperature analysis of cell killing of Bhk cells heated at temperatures in the range of 43.5^{o}c to 57.0^{o}c. *International Journal of Radiation Oncology * Biology * Physics*, 1990.

[12] D Carter, J MacFall, ST Clegg, X Wan, D Prescott, H Charles, and T Samulski. Magnetic resonance thermometry during hyperthermia for human high-grade sarcoma. *Int J Radiat Oncol Biol Phys*, 40(4):815–822, 1998.

[13] R Fernando, J Downs, D Maples, and A Ranjan. Mri/-guided monitoring of thermal dose and targeted drug delivery for cancer therapy. *Pharm Res*, 30:2709–2717, 2013.

[14] LE Gerweck. Hyperthermia in cancer therapy: The biological basis and unresolved questions. *Cancer Research*, 1985.

[15] Yuzheng Guo. *A Framework for Temperature Imaging using the Change in Backscattered Ultrasonic Signals*. PhD thesis, Washington University In St. Louis, August 2009.

[16] LR Hirsch, RJ Stafford, JA Bankson, SR Sershen, B Rivera, RE Price, JD Hazle, NJ Halas, and JL West. Nanoshell-mediated near-infrared thermal therapy of tumors under magnetic resonance guidance. *Proceedings of the National Academy of Sciences of the United States of America*, 2003.

[17] K Hynynen, A Chung, T Fjield, M Buchanan, D Daum, V Colucci, P Lopath, and F Jolesz. Feasibility of using ultrasound phased arrays for mri monitored noninvasive surgery. *UFFC, Transactions on*, 1996.

[18] L Kreyberg. Development of acute tissue damage due to cold. *Physiol Rev*, 1949.

[19] X Li, G Ghoshal, RJ Lavarello, and ML Oelze. Exploring potential mechanisms responsible for observed changes of ultrasonic backscattered energy with temperature variations. *American Association of Physicists in Medicine*, 2014.

[20] R Maass-Moreno and CA Damianou. Noninvasive temperature estimation in tissue via ultrasound echo-shifts part i analytical model. *J Acoust Soc Am*, 1996.

[21] R Maass-Moreno, CA Damianou, and NT Sanghvi. Noninvasive temperature estimation in tissue via ultrasound echo-shifts part ii in vitro study. *J Acoust Soc Am*, 1996.

[22] PM Meaney, KD Paulsen, A Hartov, and RK Crane. Microwave imaging for tissue assessment: Initial evaluation in multitarget tissue-equivalent phantoms. *Biomeg. Eng.*, 1996.

[23] J Nadobny, W Wlodarczyk, L Westhoff, J Gellermann, R Felix, and P Wust. A clinical water-coated antenna applicator for mr-controlled deep-body hyperthermia: A comparison of calculated and measured 3-d temperature data sets. *IEEE Trans Biomed Eng.*, 2005.

[24] JR Oleson and MW Dewhirst. *Hyperthermia: An Overview of Current Progress and Problems*. Year Book Medical Publishers, Inc., 1983.

[25] HH Pennes. Analysis of tissue and arterial blood temperatures in the resting human forearm. *Journal of Applied Physiology*, 1948.

[26] B Quesson, JA de Zwart, and CTW Moonen. Magnetic resonance temperature imaging for guidance of thermotherapy. *Journal of Magnetic Resonance Imaging*, 2000.

[27] SA Sapareto and WC Dewey. Thermal dose determination in cancer therapy. *International Journal of Radiation Oncology * Biology * Physics*, 1984.

[28] WL Straube and RM Arthur. Theoretical estimation of the temperature dependence of backscattered ultrasonic power for noninvasive thermometry. *Utrasound In Medicine And Biology*, 20:915–922, 1994.

[29] LM Sutherland, JA Williams, RT Padbury, DC Gotley, B Stokes, and GJ Maddern. Radiofrequency ablation of liver tumors: A systematic review. *Arch Surg*, 2006.

[30] A Szasz, O Szasz, and N Szasz. *Hyperthermia in Cancer Treatment: A Primer*, volume 3:27-59. Landes Bioscience and Springer Science+Business Media, 2006.

[31] U Techavipoo, Q Chen, and T Varghese. Ultrasonic noninvasive temperature estimation using echoshift gradient maps: Simulation results. *Ultrason Imaging*, 2005.

[32] JW Trobaugh, RM Arthur, WL Straube, and EG Moros. A simulation model for ultrasonic temperature imaging using change in backscattered energy. *Ultrasound in Medicine & Biology*, 34(2):289–298, 2008.

[33] PD Tyreus and C Diederich. Two-dimensional acoustic attenuation mapping of high temperature interstitial ultrasound lesions. *Phys Med Biol*, 2004.

[34] T Varghese, JA Zagzebski, Q Chen, U Techavipoo, G Frank, C Johnson, A Wright, and Jr. FT Lee. Ultrasound monitoring of temperature change during radiofrequency ablation: Preliminary in-vivo results. *Ultrasound in Medicine & Biology*, 2002.

[35] ED Verdonk, BK Hoffmeister, SA Wickline, and JG Miller. Anisotropy of the slope of ultrasonic attenuation in formalin fixed human myocardium. *J Acoust Soc of America*, 1996.

[36] P Wust, B Hildebrandt, G Sreenivasa, B Rau, J Gellermann, H Riess, R Felix, , and PM Schlag. Hyperthermia in combined treatment of cancer. *The lancet oncology*, 2002.

Vita

Weiyuan Zhao

Degrees B.S. China University of Mining and Technology, Electrical Engineering and Automation, June 2012

Professional Societies

Publications

December 2014

Real-Time CBE Temperature Imaging, Zhao, M.S. 2014