

University of Toronto

Undergraduate Thesis (MIE498)

Thesis

**Review and Advancements of Truncated-
Correlation Photothermal Coherence
Tomography (*TCPCT*):
Methodology, Algorithm,
Parameter Testing, and 3D Crack Imaging**

Student: P. Cui
Instructor: Professor Andreas Mandelis
Date: 2023/04/15

Table of Contents

1. Introduction:	3
➤ 1.1 Thermography and Nondestructive Testing:	3
➤ 1.2 Truncated-Correlation Photothermal Coherence Tomography (<i>TCPCT</i>):	3
➤ 1.3 Thesis Organization:	3
2. Literature Review:	4
3. Experimental Setup:	7
➤ 3.1 Measurement Setup:	7
➤ 3.2 Optical Devices:	7
4. Chirp Signals and Chirp Pulse Trains:	9
➤ 4.1 Chirp Signals:	9
➤ 4.2 Excitation / In-Phase Chirp Pulse Train:	10
➤ 4.3 Quadrature Chirp Pulse Train:	11
➤ 4.4 Some Remarks:	12
5. Post-Processing Algorithm:	14
➤ 5.1 Process the Thermal Response Data:	16
➤ 5.2 Process the Laser Excitation Data:	18
➤ 5.3 Sync the Thermal Response and Laser Excitation Data:	20
➤ 5.4 Generate the In-Phase Reference Data:	24
➤ 5.5 Generate the Quadrature Reference Data:	24
➤ 5.6 Calculate the Number of Available Slices:	24
➤ 5.7 Calculate the Amplitude and Phase Slices:	24
➤ 5.8 Save the Amplitude and Phase Slices to Disk:	30
6. Experimental Results:	31
➤ 6.1 Parameter Variation Testing:	31
➤ 6.2 3D Crack Imaging:	39
7. Summary:	40

1. Introduction:

➤ 1.1 Thermography and Nondestructive Testing:

Thermography is a technique that involves measuring the surface temperature variation of a sample after it is stimulated to determine if it has any defects. This non-invasive method of heat diffusion measurement falls under the umbrella of non-destructive testing, which is a broader field of investigation in mechanical engineering aimed at evaluating a sample's integrity without causing further damage to it.

➤ 1.2 Truncated-Correlation Photothermal Coherence Tomography (TCPCT):

In 2014, Professor Andreas Mandelis and his team at the Center for Advanced Diffusion-Wave and Photoacoustic Technologies (*CADIPT*) in the Mechanical Engineering department at the University of Toronto developed a novel thermographic imaging methodology called truncated-correlation photothermal coherence tomography (*TCPCT*) [1]. Unlike conventional thermal imaging techniques that are limited by their depth-integrated results due to the parabolic nature of thermal fields, *TCPCT* truncates the thermal response signal and slices the calculated amplitude and phase results to provide depth-localized information about the heat diffusion process. Over the past years, *TCPCT* has demonstrated significant success in subsurface defect characterization in various fields, such as in healthcare for bones and teeth [2]–[8] and in art for marquetry [7], [9], [10]. However, this technique has not yet been applied to imaging surface cracks, which is what this thesis aims to investigate. Additionally, this thesis will provide a comprehensive review of the *TCPCT* technique, introducing its current status.

➤ 1.3 Thesis Organization:

The following thesis is divided into 6 sections. After this brief introduction as **Section 1**, **Section 2** will review the previous literature works that established the foundations of this technique. **Section 3** will present the experimental setup of the entire process, followed by a description of the generation procedure of chirp pulse trains, the type of signal used by *TCPCT* to excite the sample, in **Section 4**. **Section 5** is a crucial component of this work as it presents the entire *TCPCT* data processing algorithm and the various improvements proposed by the writer of this thesis. In **Section 6**, the methodology's ability to implement crack imaging will be examined, along with the process's robustness against variations in its parameters. Also, **Section 6** will explore the technique's capability to perform 3D imaging on the crack.

2. Literature Review:

Since its initial proposal in 2014 by Kaiplavil et al. [1], the *TCPCT* technique has undergone significant developments over the past decade, finding notable applications in diverse fields such as healthcare [2]–[8] and art [7], [9], [10]. This section of the report provides a concise literature review covering the complete history of the technique's development, with an introduction to the algorithm deferred until [Section 5](#).

The idea of *TCPCT* originated from identifying a central problem in the field of thermography [1]: The propagation of thermal waves is parabolic, which causes the recorded surface temperature signal to reflect information about the defect that is integrated over the sample's depth rather than at a specific depth. To address this issue, the *TCPCT* technique applies truncation to the thermal signal and slicing to the amplitude and phase outputs to localize the depth information of the inspected sample. Four output channels were proposed in this initial paper: amplitude, initial phase, peak delay time, and zero-phase delay. It was discovered through experiments that the amplitude channel provides the greatest range and sensitivity. A 3D shape construction of the sample interior can then be obtained by passing the amplitude stack through a binarized algorithm. The experiments were carried out on various samples, including steel, bone, and skin, and all samples showed evidence that the *TCPCT* method holds significant promise in being developed as a reliable 3D defect imaging modality.

Following the first paper proposing the idea, two other works on *TCPCT* published in the same year [2], [3] offered further insights into the technique's ability to detect bone mineral loss. This research direction is noteworthy because few optical techniques offer the possibility to image hard tissues. Several additional parameters, such as thermal wave occupation index and mean planar amplitude, were developed to aid in the evaluation process. The results from *TCPCT* detection were compared to the established method of x-ray micro-computed tomography (μ CT), and the two sets of outcomes were found to follow similar trends. The two papers concluded that *TCPCT* has axial and lateral resolutions of 25 μ m and 100 μ m in this use case. [Figure 2.1](#) presents a sample slice of the result from these investigations.

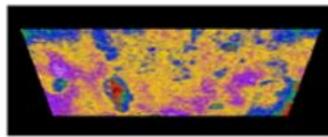


Figure 2.1. Sample Result of *TCPCT* in Bone Imaging [2]

Tavakolian et al.'s work [11] published in 2017 further clarified and improved the details of the *TCPCT* methodology, making several updates to the algorithm to elaborate its precision. The first improvement is that the enhanced modality can independently control the width of the truncated slice, providing more freedom in managing the amount of localized energy. The second improvement is that the phase channel is modified to produce the phase that corresponds to the amplitude peak rather than the minimum phase.

Additionally, the authors established a time range to ensure that the results are valid for slicing and to prevent any mixing of results due to cross-correlating the thermal signals with adjacent pulses of the reference signal.

In the more recent years, the group investigated tooth imaging as another major biomedical application for *TCPCT*. This application is more advanced than detecting mineral loss because the tooth canals are much smaller in scale than bone cavities. A series of papers have built up the argument [4]–[8], and an example *TCPCT* slice on tooth is shown in [Figure 2.2](#).

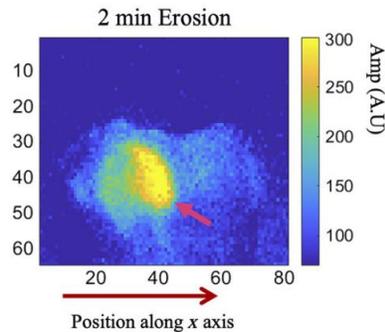


Figure 2.2. Sample Result of *TCPCT* in Tooth Imaging [5]

Another major application of *TCPCT* that the team has investigated is discovering defects in marquetry. Papers such as [7], [9], [10] have looked into this topic, and [Figure 2.3](#) presents a sample slice of result from one of these papers.

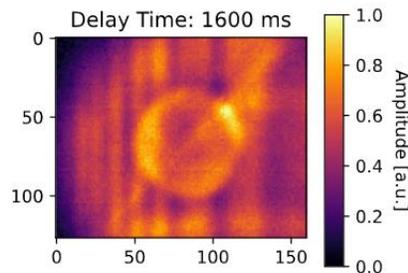


Figure 2.3. Sample Result of *TCPCT* on Marquetry Imaging [7]

The review paper by Tavakolian et al. [12] is also worthy of highlighting. In this work, a concise review on *TCPCT* and several other critical photothermal imaging techniques is offered. The other techniques include lock-in thermography, pulsed phase thermography, thermal wave radar, binary phase coding thermal coherence tomography, and chirped pulse photothermal radar.

Lastly, besides assessing the performance of *TCPCT* in various application scenarios, several other attempts have been made over the years to improve the image quality of *TCPCT* by incorporating additional signal processing steps. For instance, Zhang et al. [13] proposed an add-on modality that employed median and

Gaussian filters, polynomial regression, lower-order derivative processing, and empirical orthogonal functions to enhance *TCPCT* images. Tavakolian et al. [14] utilized a spatial-gradient-gate adaptive filter to restore features in these thermophotonic images. In another study, Rishch et al. [15] employed several computer vision techniques, including *K*-mean clustering and Canny edge detection, to enhance the boundaries of imaged defects.

3. Experimental Setup:

➤ 3.1 Measurement Setup:

Figure 3.1 displays a schematic of the experimental setup for *TCPCT*.

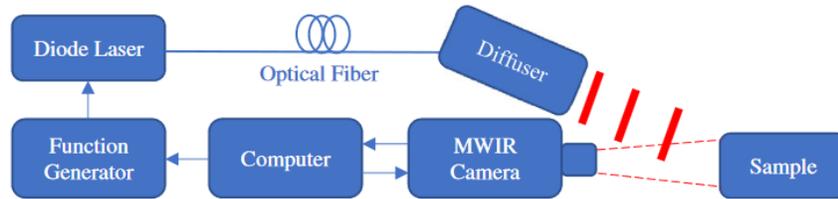


Figure 3.1. Experimental Setup Schematic for *TCPCT* [7]

The testing process starts with the experimenter designing the waveform used to control the laser, which is done by entering the required signal parameters into a self-developed computer program created for laser control. The computer then transmits the signal parameters to a function generator (Keysight 33500B, Santa Rosa, CA) to form the waveform. After the waveform is created, it is sent to the laser driver (PCO-6131, Directed Energy, Fort Collins, CO) for activation. Depending on the activation waveform, the 808-*nm* pulsed diode laser (Jenoptic JOLD-120-QPXF-2P) is either turned on or off. Meanwhile, the actual activation voltage used by the laser driver is sampled by another self-developed high-speed data acquisition module and stored in the computer as the excitation signal. When activated, the laser emits powerful infrared radiation that passes through an optical fiber and finally reaches the sample surface, resulting in heat deposition and the initiation of surface temperature increase and decrease. The infrared camera (A6700sc, FLIR, Wilsonville, OR) captures this temperature change and transmits it back to the computer for storage. Therefore, at the end of the experiment, the experimenter will have two data files: one from the laser driver representing the excitation signal, and the other from the camera software representing the thermal response.

➤ 3.2 Optical Devices:

It should be also noted that this lab setup incorporates several optical devices that are of great importance. One is the collimator (F220SMA-B, Thorlabs, Inc., Newton, NJ) that is attached at the end of the optical fiber to reduce the spread and divergence of the laser beam. Laser diodes have shorter optical cavities and therefore produce less collimated beams compared to gas and crystal lasers, making a laser collimator necessary for optimal performance in the *TCPCT* experimental setup. In Figure 3.2, a schematic drawing of a laser collimator is illustrated, which consists of three main components: a casing, a molded aspheric lens, and a plug adapter. The lens is responsible for redirecting the laser beam to achieve improved collimation, while the plug adapter facilitates connection to an optical fiber [16], [17].

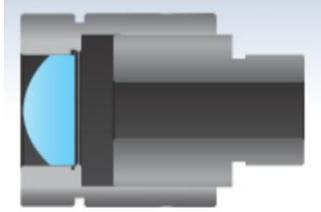


Figure 3.2. Laser Collimator Schematic [16]

Another device utilized in the experiment is a diffuser (circle pattern ED1-C20, Thorlabs, Inc., Newton, NJ), placed closer to the sample surface. The diffuser's purpose is to scatter the beam, spreading it out to cover a larger area of the tested surface and to distribute the laser intensity more evenly. However, as the laser power remains constant, scattering is expected to decrease the level of laser intensity per unit area significantly. For instance, experiments show that when the laser emits a short pulse of 100 *ms*, the maximum temperature rise on the sample surface is roughly 70 °C and 4 °C for cases with and without diffusers, respectively.

Lastly, in addition to the two optical components altering the laser trajectory, the camera lens is another piece of instrument that significantly influences the experiment's outcome. Similar to using microscopes, changing the lens with varying focal lengths alters the magnification of the camera and its field of vision.

4. Chirp Signals and Chirp Pulse Trains:

The *TCPCT* technique employs chirp pulse trains to stimulate the sample. As such, this section will outline the definition and construction process for both chirp signals and chirp pulse trains.

➤ 4.1 Chirp Signals:

Chirp signals refer to signals whose frequencies vary over time. These signals can be categorized as up-chirps, which have increasing frequencies, or down-chirps, which have decreasing frequencies. Frequency modulation can be applied to either the instantaneous linear or angular frequency of the signal, as both frequencies are always related to each other by a factor of 2π . Although there are various types of chirp signals (e.g., linear, exponential, hyperbolic), linear chirp signals are the most commonly used. This is because they have a simple and fundamental structure, with a linear increase or decrease in frequency over time [18]. Figures 4.1a and 4.1b demonstrate the waveform of such signals.

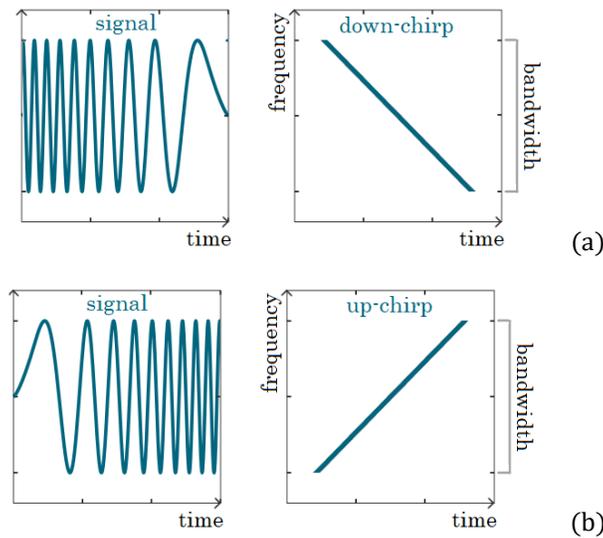


Figure 4.1. Linear Sinusoidal (a) Up-Chirp and (b) Down-Chirp Signals [19]

Mathematically, linearly modulated chirp signals, which have modulated linear frequencies, can be represented by the following equations:

$$f(t) = f_1 + ct$$

$$c = \frac{f_2 - f_1}{T}$$

As for those signals that have modulated angular frequencies, the corresponding equations are:

$$\omega(t) = \omega_1 + kt$$

$$k = \frac{\omega_2 - \omega_1}{T}$$

In these equations, f_1 and f_2 are the starting and ending frequencies of the chirp signal, ω_1 and ω_2 are the starting and ending angular frequencies, T is the chirp period, c is the linear frequency chirp rate, and k is the angular frequency chirp rate. It is worth noting that c and k are also sometimes referred to as the linear and angular sweep rates of the chirp signal, respectively [18].

➤ **4.2 Excitation / In-Phase Chirp Pulse Train:**

In order to create a chirp pulse train, it is crucial to first determine the phase of the corresponding chirp signal. This can be achieved by integrating the angular frequency function $\omega(t)$ over time:

$$\begin{aligned}\Delta\phi(t) &= \int_0^t \omega(\tau) d\tau \\ \Delta\phi(t) &= \int_0^t (\omega_1 + k\tau) d\tau \\ \Delta\phi(t) &= (\omega_1\tau + k\tau^2)_0^t \\ \Delta\phi(t) &= \omega_1 t + \frac{1}{2} k t^2 \\ \phi(t) - \phi_1 &= \omega_1 t + \frac{1}{2} k t^2 \\ \phi(t) &= \phi_1 + \omega_1 t + \frac{1}{2} k t^2\end{aligned}$$

Here, ϕ_1 is the starting phase of the chirp signal. Note that since the expression for $\phi(t)$ has a second-order relationship with respect to time, linearly modulated chirp signals are also sometimes referred to in literature as quadratic-phase signals.

Once the expression for $\phi(t)$ of the linearly modulated chirp signal is obtained, a pulsed chirp can be synthesized by generating a pulse each time the phase reaches a certain magnitude over 2π . For the excitation data of *TCPCT*, this magnitude is set to be $\pi/2$. Mathematically, the positions of the pulses are solutions to the following equation:

$$\phi(t) = \omega_1 t + \frac{1}{2} k t^2 = 2m\pi + \frac{\pi}{2}$$

where m is an integer starting from 0. In other words, if a pulse train of p_d pulses is needed, then the locations of the pulses on the time scale are given by:

$$\phi(t_{p,0^\circ,m}) = \omega_1 t_{p,0^\circ,m} + \frac{1}{2} k t_{p,0^\circ,m}^2 = 2m\pi + \frac{\pi}{2} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

In this context, the subscript p denotes pulses, 0° refers to in-phase data, and m represents the index of pulses within the entire pulse train (beginning from 0). It is worth noting that this excitation pulse train is also

known as the in-phase pulse train since there is no time delay between the pulses and the thermal excitation peaks, meaning that they are always synchronized with each other.

Then, to obtain a direct representation of $t_{p,0^\circ,m}$, solving the equation above quadratically to yield:

$$\frac{1}{2}kt_{p,0^\circ,m}^2 + \omega_1 t_{p,0^\circ,m} - \left(2m\pi + \frac{\pi}{2}\right) = 0 \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,m} = \frac{-\omega_1 \pm \sqrt{\omega_1^2 - 4 * \frac{1}{2}k * \left(-\left(2m\pi + \frac{\pi}{2}\right)\right)}}{2 * \frac{1}{2}k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,0^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + (4mk\pi + k\pi)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,0^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + k\pi(4m + 1)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

Finally, using the expression of $t_{p,0^\circ,m}$, the chirp pulse train that controls the laser to excite the sample can be written as follows:

$$P(t) = P_{0^\circ}(t) = \sum_{m=0}^{p_d-1} \delta(t - t_{p,0^\circ,m})$$

where:

$$t_{p,0^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + k\pi(4m + 1)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$k = \frac{\omega_2 - \omega_1}{T}$$

➤ 4.3 Quadrature Chirp Pulse Train:

In the field of thermography, particularly in the lock-in thermography process, a second set of signals known as the quadrature signal is utilized in conjunction with the in-phase signal. The quadrature signal is delayed by 1/4 of a cycle in comparison to the in-phase signal, and together, they are used to generate the amplitude and phase outcomes. This same approach is also employed in *TCPCT*. To introduce a 90° delay, the quadrature pulse train's pulses are identified when the chirp signal's phase is 180° instead of 90°. As a result, the analytical results for the locations of the pulses in this additional pulse train are:

$$\phi(t_{p,90^\circ,m}) = \omega_1 t_{p,90^\circ,m} + \frac{1}{2}kt_{p,90^\circ,m}^2 = 2m\pi + \pi \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$\frac{1}{2}kt_{p,90^\circ,m}^2 + \omega_1 t_{p,90^\circ,m} - (2m\pi + \pi) = 0 \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,90^\circ,m} = \frac{-\omega_1 \pm \sqrt{\omega_1^2 - 4 * \frac{1}{2}k * (-(2m\pi + \pi))}}{2 * \frac{1}{2}k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,90^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + (4mk\pi + 2k\pi)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

$$t_{p,90^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + k\pi(4m + 2)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

Similar to the in-phase calculations, the quadrature pulse train can be expressed as follows:

$$P_{90^\circ}(t) = \sum_{m=0}^{p_d-1} \delta(t - t_{p,90^\circ,m})$$

where:

$$t_{p,90^\circ,m} = \frac{-\omega_1 + \sqrt{\omega_1^2 + k\pi(4m + 2)}}{k} \quad (m = 0, 1, 2, \dots, (p_d - 1))$$

➤ 4.4 Some Remarks:

To conclude this section, it is first worth emphasizing that the excitation / in-phase chirp pulse train is already present after the measurement, given that it is utilized to stimulate the inspected sample. On the other hand, the quadrature pulse train is introduced solely during the post-processing phase to aid in the generation of amplitude and phase slices of heat wave propagation.

Another characteristic of chirp pulse trains that is worth noting is that the interval between adjacent pulses for an up-chirp pulse train always decreases. This property is related to the phase equation discussed earlier:

$$\phi(t) = \omega_1 t + \frac{1}{2}kt^2 = 2m\pi + \frac{\pi}{2}$$

For an up-chirp pulse train, where $\omega_2 > \omega_1$, the chirp rate $k > 0$. Therefore, the phase expression $\phi(t)$ is a concave upward quadratic function. Then, since the coefficients in front of t^2 and t are both positive, the axis of symmetry of this parabola is in the negative half-plane for x . Additionally, this expression always passes through the point $(t = 0, \phi(t) = 0)$. Thus, the graph of $\phi(t)$ versus t for an up-chirp pulse train appears as shown in [Figure 4.2](#). In the positive half-plane for x , $\phi(t)$ rises at an increasing rate, while the vertical axis values that the function is trying to intercept, $(2m\pi + \pi/2)$, has a constant rate of change. Consequently, the x values associated with larger $\phi(x)$ values will be more closely spaced, meaning that later pulses in an up-chirp pulse train are closer together.

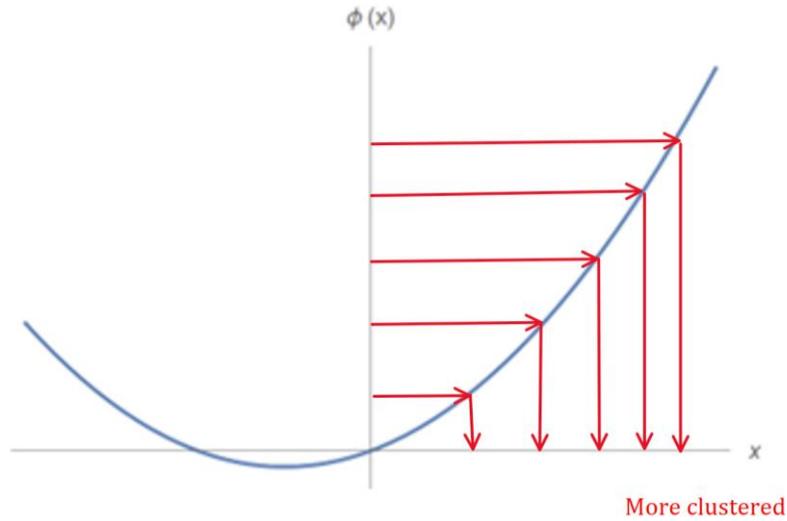


Figure 4.2. Solution Distribution of the Phase Equation for an Up-Chirp Pulse Train

An inverse conclusion can be given for down-chirp pulse trains. As shown in [Figure 4.3](#), the time interval between consecutive pulses becomes larger. However, this case has some additional restrictions coming from the fact that the phase first increases and then decreases when $t > 0$. This indicates that only between $\phi(x) = 0$ and $\phi(x) = \text{peak value of the phase curve}$ will there be solutions of x . Hence, when studying *TCPCT*, to avoid the problems of encountering no appropriate pulse locations, only up-chirp pulse trains are used. In addition, up-chirp pulse trains have the advantage that the earlier pulses are given more time to cool down, so their effects on the subsequent pulses are smaller than down-chirp pulse trains.

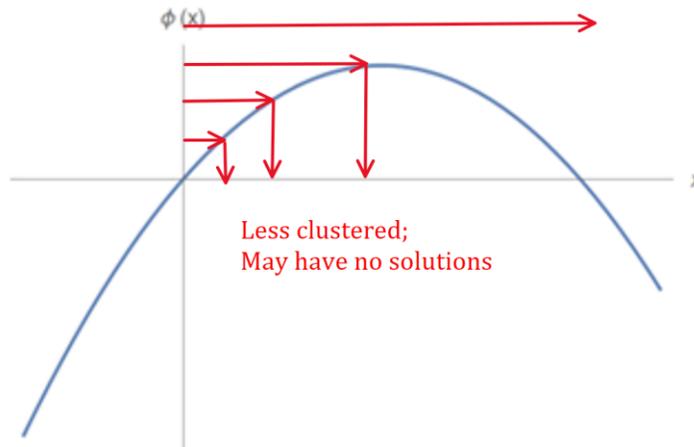


Figure 4.3. Solutions of the Phase Equation for a Down-Chirp Pulse Train

5. Post-Processing Algorithm:

In this section, the *TCPCT* post-processing algorithm will be introduced in detail. Besides, the improvements on this process implemented by the author of this thesis will be highlighted. Overall, the primary objective of this algorithm is to use the recorded data of laser excitation and thermal response and the additional operations of truncation and slicing to produce amplitude and phase slices that reflect the depth-localized heat progression inside the tested specimen.

Before delving into the algorithm itself, it is crucial to clarify that the term *TCPCT* is actually a misnomer in most literature. There are, in fact two distinct versions of the algorithm known as *TCPCT*. The *TCPCT* algorithm as presented by Welch et al. [7] is the sole one that applies truncation followed by cross-correlation to the signals. On the other hand, in other publications, whether labeled as *TCPCT* or *eTCPCT*, the algorithm actually performs cross-correlation first and then slices the results to generate the final images. Despite this distinction, both methods have been proven to possess the ability to provide depth-localized defect characterization. However, it is noteworthy that the second method is more commonly studied than the first, owing to the considerably higher number of calculations involved in the first method, resulting in longer processing times, as elucidated in subsequent sections.

To avoid confusion between the two versions of the *TCPCT* algorithm, this thesis will henceforth refer to the version that uses first cross-correlation and then slicing as *eTCPCT*, which has been extensively studied. The version that applies first truncation and then cross-correlation to obtain the desired results, which is currently still being developed, will be referred to as *TCPCT*.

A flowchart of the entire post-processing algorithm that applies to both *eTCPCT* and *TCPCT* is presented in [Figure 5.1](#) on the next page. Both post-processing processes can be summarized into 8 steps, with the only difference between the two methods occurring in [Step 7](#).

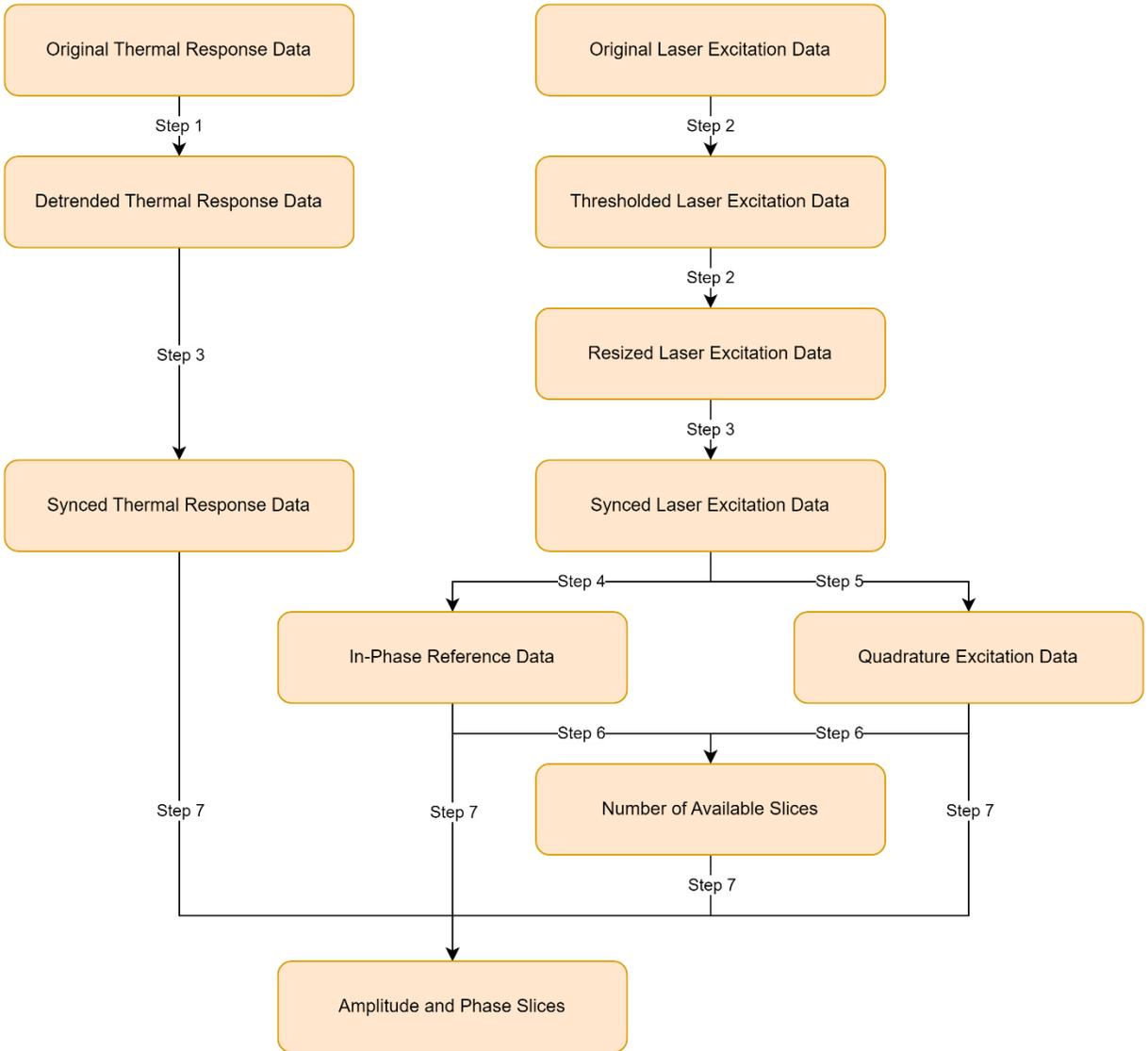


Figure 5.1. Post-Processing Flowchart for Applicable to Both *eTCPCT* and *TCPCT*

➤ 5.1 Process the Thermal Response Data:

(1) Import:

To initiate the post-processing procedure, the thermal camera recording frames are read sequentially by the computer to construct a 3D matrix that comprises the complete thermal response data. Each frame represents the temperature distribution of the examined surface at a specific time point. This 3D matrix can be denoted as $T_{ori}(x, y, f\#)$, where T_{ori} indicates that it is an original, unprocessed temperature signal, (x, y) specifies that the signal value is dependent on the corresponding pixel location, and $f\#$ indicates that the signal value varies with each frame captured. The dimensions of the resulting thermal response matrix can then be calculated by multiplying the full frame height, full frame width, and the total number of frames. In equation form:

$$[\text{Size of Thermal Response Data}] = [\text{Full Frame Height}] * [\text{Full Frame Width}] * [\text{Total Frame Number}]$$

Figure 5.2 depicts an example frame from a thermal camera recording, and Figure 5.3 illustrates an example thermal data for a specific pixel point across all recorded frames in a sample dataset.

In the remainder of this paper, a colon (:) will be used to indicate the case in which an entire independent dimension of a signal is selected. Therefore, using this notation, a thermal camera frame can be written as $T_{ori}(:, :, f\#)$, and a specific pixel's thermal response can be written as $T_{ori}(x, y, :)$.

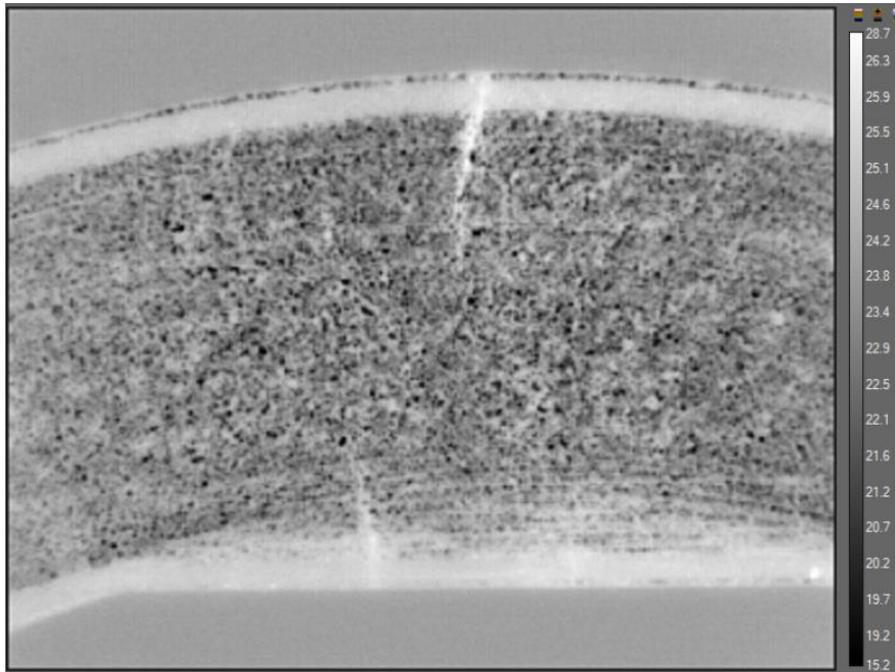


Figure 5.2. Original Surface Temperature Profile from a Sample Dataset at a Specific Time Stamp (i.e., an Example of $T_{ori}(:, :, f\#)$)

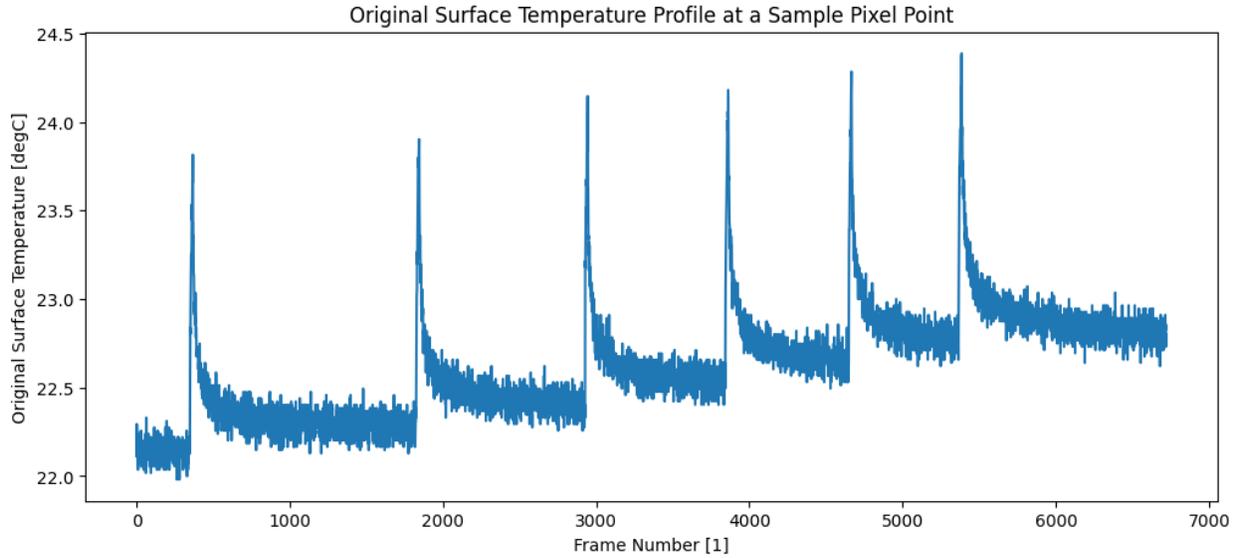


Figure 5.3. Original Surface Temperature Profile from a Sample Dataset at a Specific Pixel Point (i.e., an Example of $T_{ori}(x, y, :)$)

(2) Detrend:

Although the laser pulses that excite the inspected sample occur sequentially in time, in both *eTCPCT* and *TCPCT*, they are treated as independent events. To avoid the impact of thermal drifting caused by the repetitive heating of the sample surface, detrending the thermal response data is essential. Failure to do so result in different pulses being on different magnitudes due to the leftover heat accumulation from earlier pulses. The detrended temperature signal of that presented in Figure 5.3 is shown in Figure 5.4. Following the notations for the original thermal response, the detrended thermal response is denoted as $T_{det}(x, y, f\#)$.

Readers of the report may raise questions about the necessity of multiple excitation pulses when they are treated as unrelated. To address this, let us revisit the discussions on chirp pulse trains mentioned in Section 4.3. It is important to recall that we need to generate a quadrature pulse train, which lags by 90° compared to the excitation / in-phase chirp pulse train. As explained in Section 4.4, up-chirp pulse trains exhibit pulses that gradually cluster together. Consequently, when creating the quadrature pulse train, different quadrature pulses are shifted by varying amounts of time relative to their corresponding in-phase pulses. This results in a new pulse train that is not a simple time shift of the entire in-phase pulse train. Therefore, the presence of multiple excitation pulses, determined by the starting frequency, ending frequency, and duration of the chirp signals, remains a crucial aspect of this algorithm.

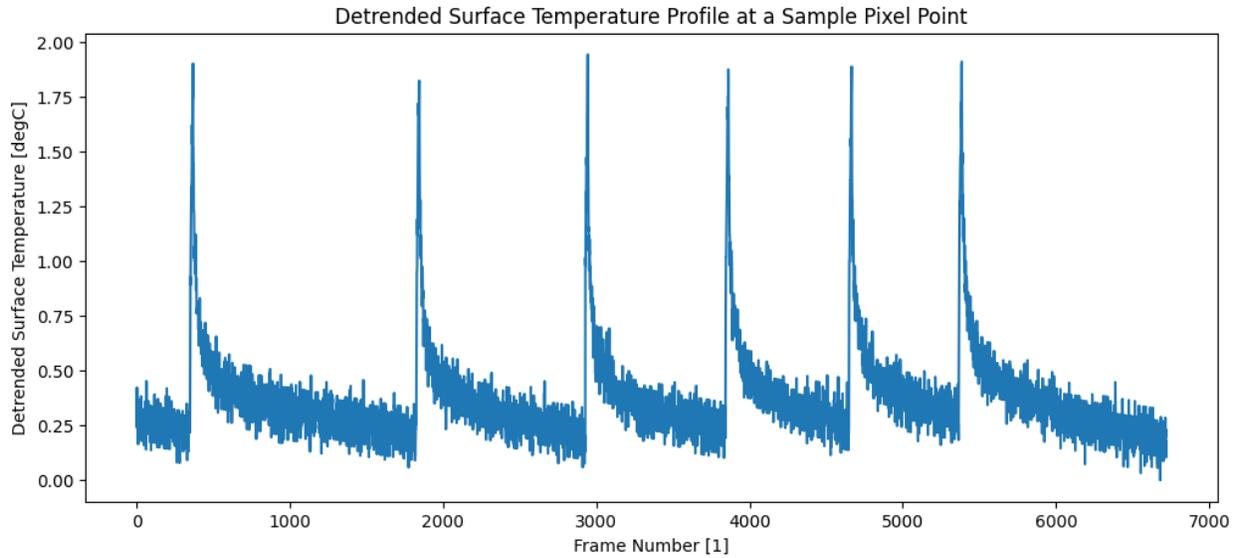


Figure 5.4. Detrended Surface Temperature Profile at a Specific Pixel Point
(i.e., an Example of $T_{det}(x, y, :)$) (Produced Using the Same Dataset as the Prior Figures)

➤ 5.2 Process the Laser Excitation Data:

(1) Import:

After loading the thermal response data, the laser excitation data is also imported. The excitation data contains some noise since as mentioned in [Section 3](#), it is a resampling of the exact excitation amplitude used by the laser driver to activate the laser. [Figure 5.5](#) shows an example sequence of the laser excitation data from the same dataset as [Figure 5.3](#).

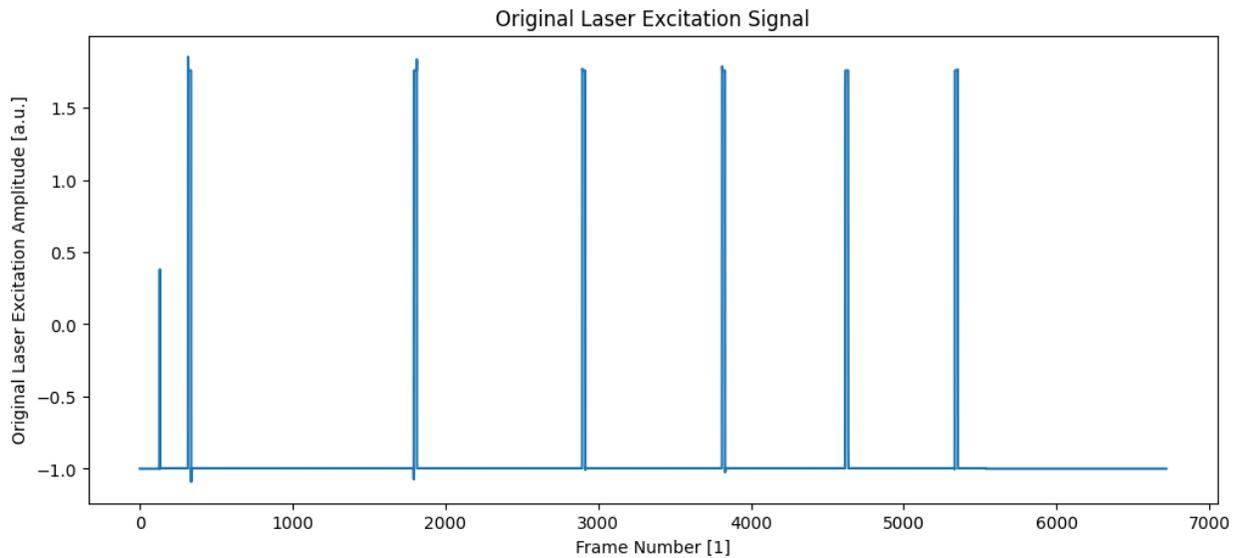


Figure 5.5. Original Laser Excitation Data
(Produced Using the Same Dataset as the Prior Figures)

(2) Correct:

Some corrections is then applied on the excitation data imported above. The desired output is a 0/1 binary array with 0 and 1 mapping to the *ON* and *OFF* entries of the laser, respectively. To achieve this, a threshold is applied to the original excitation data, and the trigger pulse that does not activate the laser, found at the beginning of the data, is removed. **Figure 5.6** displays a sample result obtained after processing the data from **Figure 5.5**.

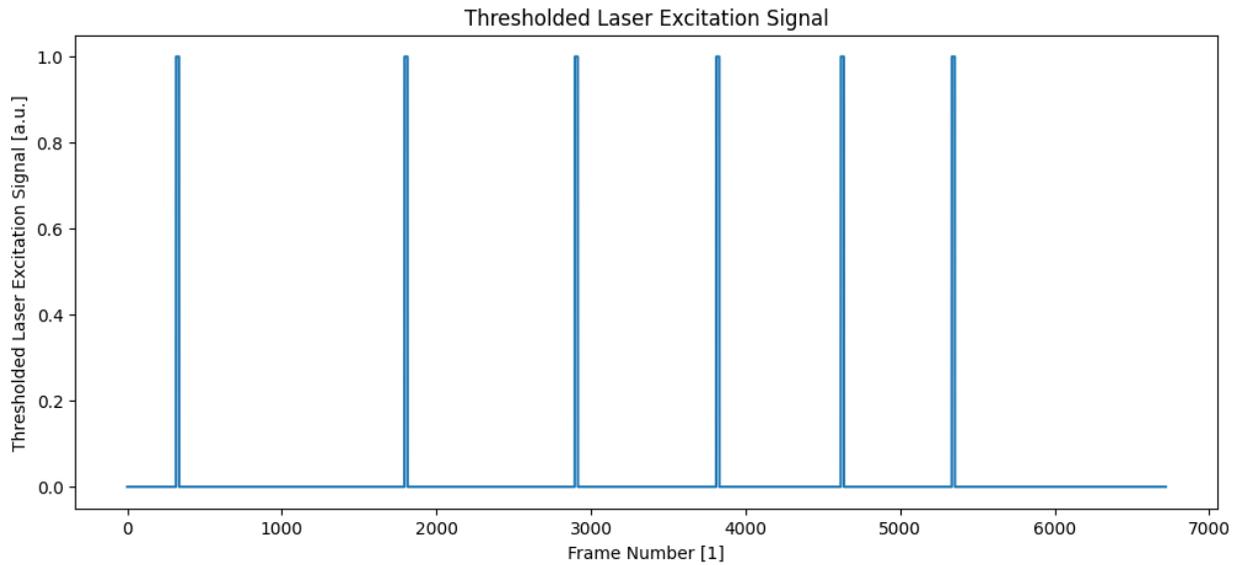


Figure 5.6. Thresholded Laser Excitation Data
(Produced Using the Same Dataset as the Prior Figures)

Given that the duration of the pulses and the frame rate of the camera are known inputs for the experiment, a quick sanity check to verify the validity of the data and its processing is to sum up the entire excitation data array and compare it to the expected number of *ON* entries for all pulses. The expected number can be calculated as the product of the pulse number, frame rate, and pulse duration, represented as:

$$[\text{Total Number of } ON \text{ Entries for All Pulses}] = [\text{Pulse Number}] * [\text{Frame Rate}] * [\text{Pulse Duration}]$$

(3) Resize:

It should be further noted that the length of the excitation data may not always match the number of frames in the recording due to dropped frames in the thermal camera recording or some other reasons. To address this issue, a resizing function was developed to adjust the length of the excitation data to match the number of frames precisely. The resizing function uses the number of camera frames as a reference and modifies the length of the excitation data accordingly. If the reference data is longer than the number of frames, then the function truncates the last few entries of the data. If the reference data is shorter than the number of frames, then a few zero entries is added to the excitation data's end.

➤ 5.3 Sync the Thermal Response and Laser Excitation Data:

The *TCPCT* algorithm relies on precise synchronization between the thermal response and laser excitation signals. Only when these two signals are perfectly aligned can the dataset be meaningfully processed further.

After processing the laser excitation signal as shown in the previous step, it is straightforward to correspond its elements to the laser activation status: a value of 1 indicates that the laser is *ON*, while a value of 0 indicates that the laser is *OFF*. However, identifying the pulses in the thermal response signal is a challenging task. There is a significant level of noise present in the measured thermal data (refer back to [Figures 5.3 and 5.4](#)), and the shape of the measured thermal decay varies across different experiments during to distinct pulse inputs. The only clear characteristic that can help in identifying the pulses is that the thermal response signal peaks at the end of each pulse, when the heat deposition on the sample surface is completed.

Therefore, to overcome these challenges, the thesis writer improved upon the previous algorithm and devised a robust peak identification method using Python's SciPy module with customized arguments tailored to the shape of the thermal decay signals. Specifically, three key additions were made to enhance the peak identification process:

1. In order to reduce the impact of noise, the averaged temperature signal across slices, instead of the thermal response signal at any particular pixel, was used as the representative thermal response data for signal alignment.
2. An argument for peak prominence was added to the implementation of SciPy's "find_peak" command to obtain a short but reliable initial list of peak candidates, which speeds up the subsequent refined search of peaks.
3. A while loop was constructed to gradually increase the required peak width for the most prominent peak candidates.

After implementing these three improvements, the code was tested on all available datasets, and no incorrect data syncing was observed. This suggests that the peak search argument is both robust and effective.

Additionally, to visually demonstrate the effectiveness of the updates, [Figures 5.7 and 5.8](#) presented some sample results from before and after the data alignment process. These figures further confirmed the validity of the updated peak identification method.

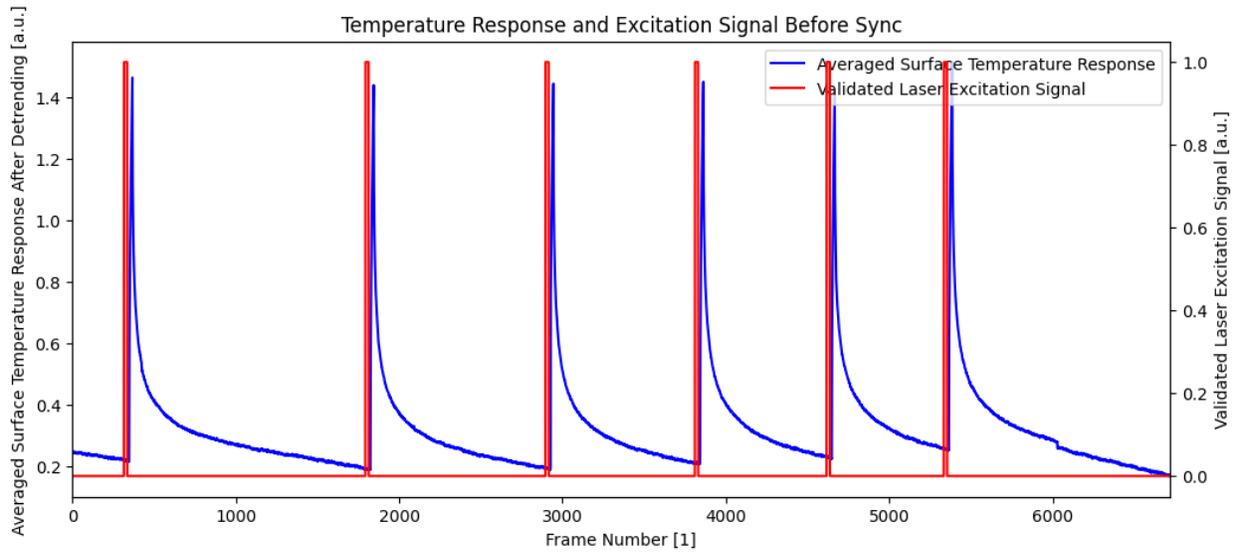


Figure 5.7. Temperature Response and Laser Excitation Signals Before Sync
 (Produced Using the Same Dataset as the Prior Figures)

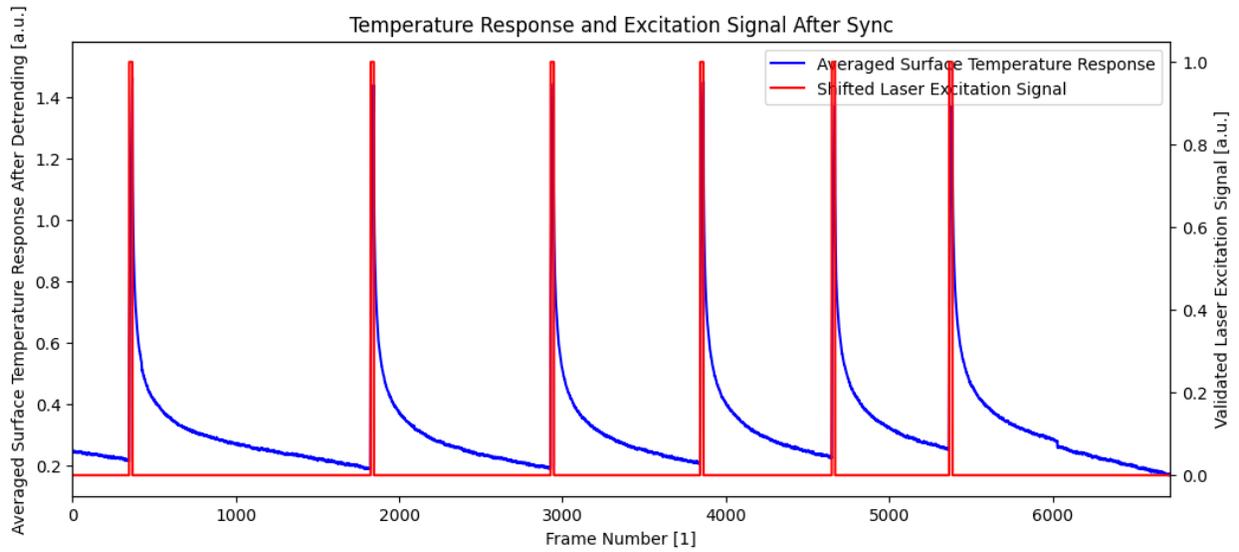


Figure 5.8. Temperature Response and Laser Excitation Signals After Sync
 (Produced Using the Same Dataset as the Prior Figures)

The thesis author would like to emphasize the rationale behind using peak prominence as the first criterion for identifying peak candidates. Peak prominence is a measure of how distinct a peak is in terms of its intrinsic height and relative location among other peaks. Figure 5.9 presents a schematic of a series of peaks, with their prominence illustrated as the height of the colored region below each peak. For instance, the prominence of peak 6 is calculated as the height difference between points 6 and d, while the prominence of peak 8 is the height difference between points 8 and f [20].

The determination of peak prominence involves a 4-step process [20]:

1. Extend a horizontal line from the peak towards the left and right directions until it encounters a higher peak or reaches the boundaries of the graph.
2. Identify the minimum value of the signal within each interval formed by the peak and the endpoints obtained from Step 1.
3. Select the higher of the two interval minima as the reference level.
4. Calculate the peak prominence as the difference in height between the larger minimum from Step 3 and the peak itself.

Essentially, peak prominence identifies the importance of a peak relative to its adjacent environment where it stands out the most. Referring back to Figure 5.9, peak 4 exhibits the highest prominence among the group of peaks 3 and 4. However, beyond these two peaks, in a larger vicinity, there are other peaks, such as peaks 2 and 6, that surpass peak 4 in height. Hence, peak 4 only dominates within the range of peaks 3 and 4, and the reference level chosen for measuring its prominence is the base level of the curve for these two peaks.

In our present case of peak search on thermal response data, peak prominence can be demonstrated to have great importance, as seen in Figure 5.10. The differences in peak prominences between a genuine peak resulting from thermal excitation (e.g., peak A_1) and a spurious peak caused by noise during thermal decay (e.g., peak A_2) are clearly evident. For peak A_1 , a horizontal line passing through it intersects the graph again at points B_1 and C_1 , where B_1 is the result of reaching the graph boundary, and C_1 is due to reaching the thermal rise induced by another pulse. Then, the larger minimum found in the two intervals (B_1, A_1) and (A_1, C_1) is located at D_1 . Therefore, the height between A_1 and D_1 represents the peak prominence for A_1 . On the other hand, for peak A_2 , the same process is followed, but the horizontal line extended to the left abruptly stops at B_2 due to the higher peak of A_1 located in close proximity. As a result, the selected minimum corresponding to B_2 is just D_2 , the small valley to its left. This significantly decreases the prominence of A_2 , leading to the conclusion that peak prominence is a reliable parameter for identifying genuine peaks corresponding to thermal pulse excitations.

As further evidence to support this argument, Figure 5.11 displays a series of peak search results obtained using different referencing parameters, including peak distance, prominence, width, and threshold. Clearly, the results obtained by searching based on peak prominence are the most appropriate and reliable, reinforcing the significance of peak prominence as a robust parameter for peak identifications.

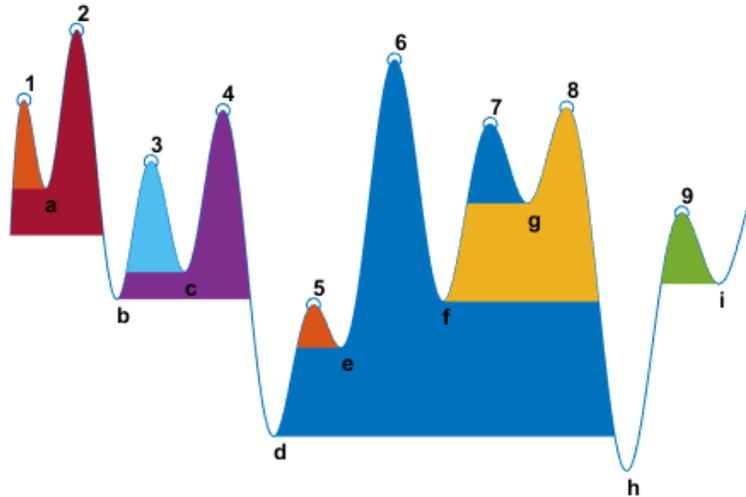


Figure 5.9. A Sample Peak Series and Their Prominences [20]

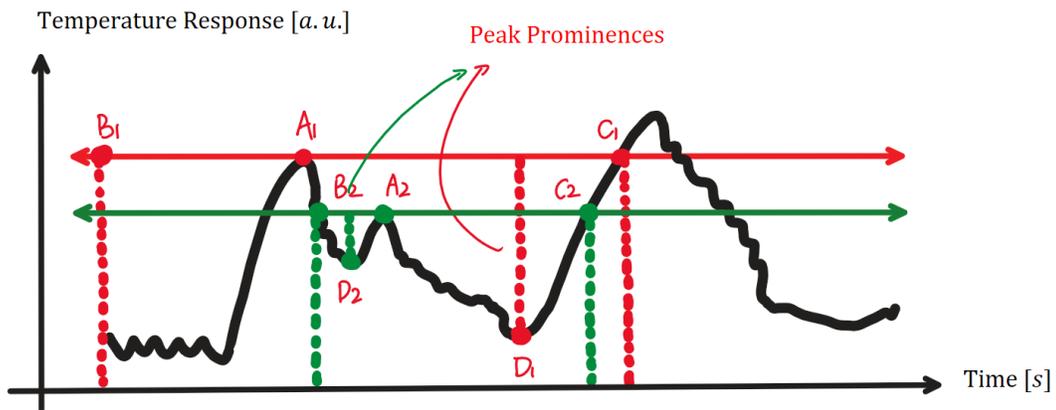


Figure 5.10. Peak Prominence Search Applied on Thermal Response Data

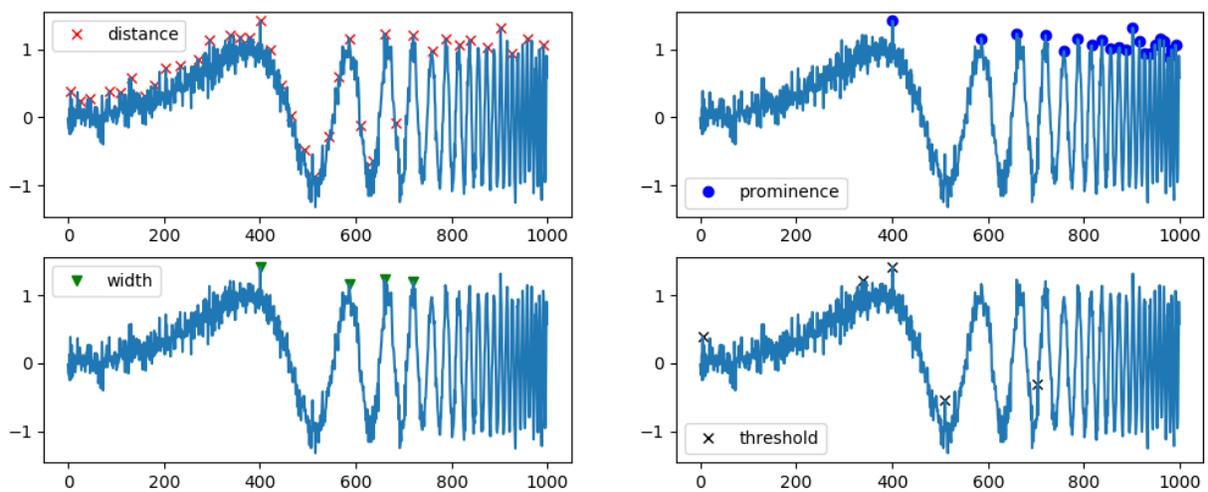


Figure 5.11. Differences Peak Search Results Using Distance, Prominence, Width, and Threshold as the Selection Parameter [21]

➤ 5.4 Generate the In-Phase Reference Data:

After obtaining a properly aligned pair of thermal response and laser excitation data, the next step is to generate an in-phase reference data. The in-phase reference data consists of a pulse train with pulses located at the same positions as those in the laser excitation data. However, the width of the pulses is adjusted to match the slice width used in the subsequent truncation or slicing steps for slice production.

➤ 5.5 Generate the Quadrature Reference Data:

In [Section 4](#), it was previously explained that quadrature signals are signals that lag by 90° compared to the in-phase signals. This phase difference is taken into account when using the phase equation to determine the location of pulses in chirp pulse trains. [Section 4](#) also provided an analytical method for generating quadrature chirp pulse trains. However, for the sake of computational efficiency, a slight modification has been made in the actual calculations. Specifically, the quadrature pulses are lagged by exactly $1/4$ of the distance between the current pulse and the next pulse. Then, similar to the construction of the in-phase reference signal, the width of the *ON* portion of the data is determined based on the width of the slices used in the subsequent steps of the process.

➤ 5.6 Calculate the Number of Available Slices:

➤ 5.7 Calculate the Amplitude and Phase Slices:

(1) Introduce:

These two steps of the algorithm are presented together since understanding the process for calculating amplitude and phase slices ([Step 7](#)) is crucial in determining the appropriate time range for making these slices ([Step 6](#)). It is also important to note that [Step 7](#) marks a significant departure point between the *eTCPCT* and *TCPCT* algorithms. Therefore, in this section, the method for calculating the amplitude and phase slices in *eTCPCT* is explained first, followed by that used in *TCPCT*. Then, the similarities in their calculations are identified, leading to the conclusion that both algorithms have the same time range for producing the desired slices.

In *eTCPCT*, the amplitude and phase slices are obtained through a sequence of steps that involve cross-correlation and slicing. A flowchart illustrating the entire process is provided as [Figure 5.14](#). The detailed steps are as follows:

1. (Cross-Correlation) The in-phase and quadrature reference signals are cross-correlated with the entire thermal response data to produce the in-phase and quadrature cross-correlation results.
2. These two results are combined based on corresponding terms to yield the amplitude and phase sequences.
3. (Slicing) Starting from the zero cross-correlation delay time, the positive axis of the delay time is traversed, and the amplitude and phase results are sliced at every multiple of the slice width.
4. This process generates a sequence of *eTCPCT* amplitude and phase slices.

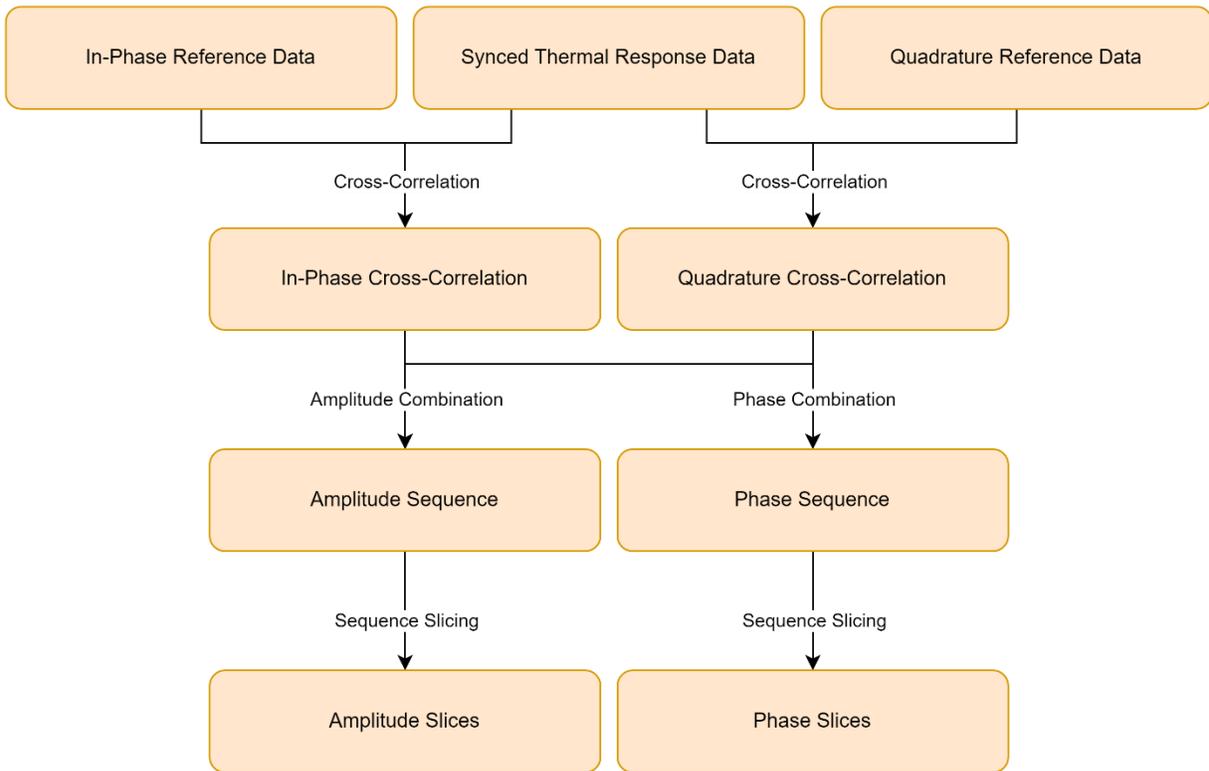


Figure 5.14. *eTCPCT* Slice Calculation Procedure

The processing steps for *TCPCT* are slightly different, with the first being truncation and then followed by cross-correlation. A flowchart showing the entire process is presented as [Figure 5.15](#). The details of the process are also listed as follows:

1. (Truncation) The in-phase reference signal is first delayed by a certain time (several multiples of the slice width) and then multiplied with the thermal response signal to truncate a portion of the thermal decay for specific focus. The selected section of the thermal signal has a length equal to the pulse width of the in-phase reference data, which is also equal to the slice width.
2. (Cross-correlation) The truncated part of the thermal signal is then cross-correlated with both the in-phase and quadrature reference signals to generate two sequences of cross-correlation results. Notably, due to the sparse nature of both the truncated thermal signal and the reference signal (either in-phase or quadrature), these cross-correlation outcomes are also sparse.
3. To avoid one value dominating the other, only entries whose in-phase and quadrature cross-correlation values are both nonzero are considered when combining these two sequences into amplitude and phase results. Other entries are set to zero.
4. The maxima across the entire amplitude and phase sequences are then found to form the images, resulting in a pair of amplitude and phase slices.
5. To generate a stack of amplitude and phase images, the delay time applied on the in-phase reference signal in Step 1 of the process is changed to different multiples of the slice width. This truncates different parts of the thermal decay, yielding a series of thermal slices that are used to form the stack of amplitude and phase images.

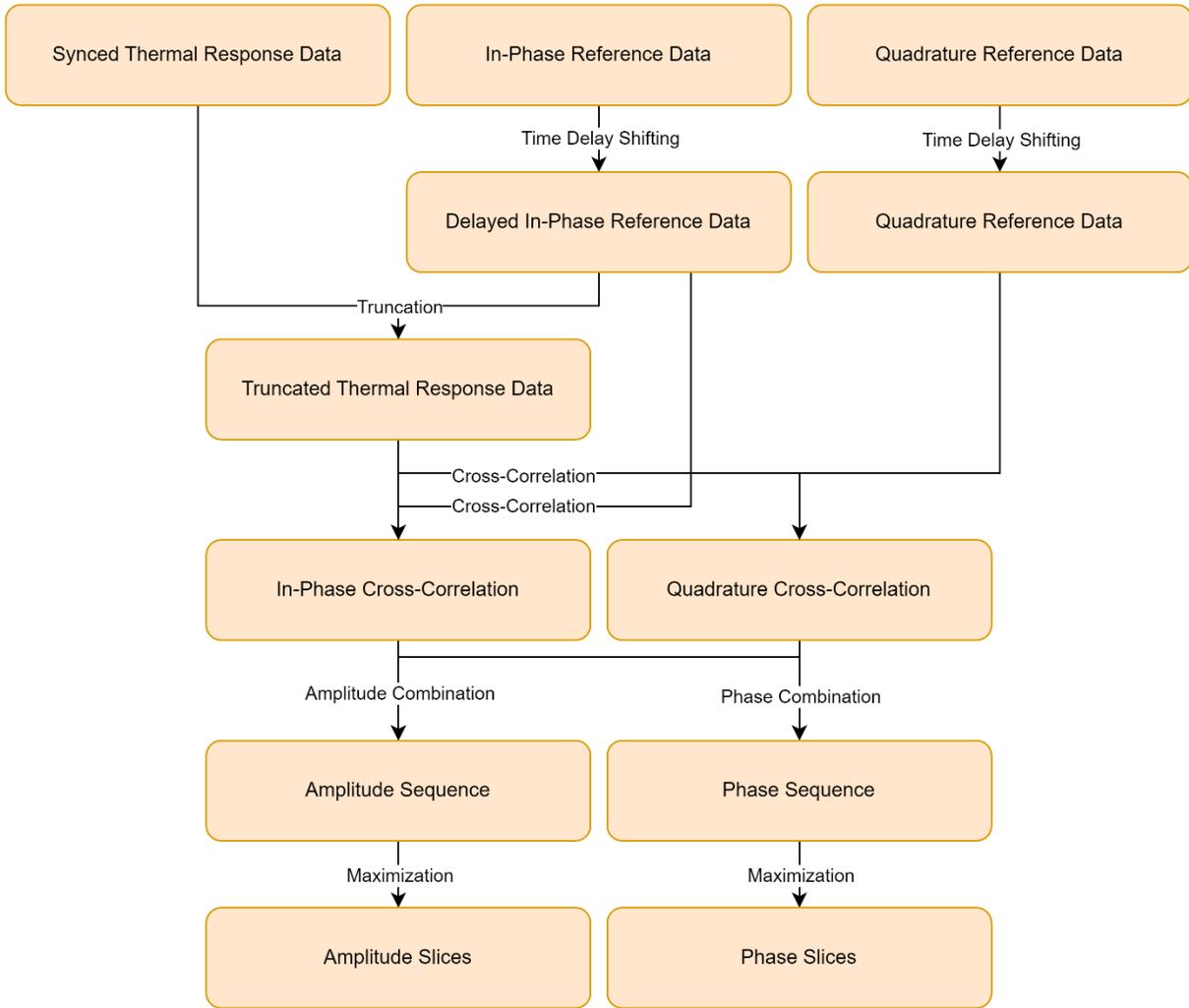


Figure 5.15. TCPCT Slice Calculation Procedure

(2) Compare:

A comparison between the two algorithms reveals that the *eTCPCT* algorithm, for all its slices, only requires one cross-correlation per pixel to generate results. On the other hand, the *TCPCT* algorithm involves different truncations of the thermal signal, and each truncation is associated with one cross-correlation between the delayed reference signal and the truncated thermal signal. As a result, the computational workload of *TCPCT* is significantly heavier than *eTCPCT*.

During the initial stages of this thesis project, the lab had already developed software capable of calculating amplitude and phase slices using both methods. However, the computational speed of *TCPCT* was found to be too slow, taking over 12 hours to process 120 slices. This substantial time requirement poses a significant disadvantage in terms of further development of *TCPCT*. Fortunately, in the next section, the author of this thesis will introduce some successful efforts to speed up the process and bring it within a reasonable timeframe.

Another important consideration is the valid time range for both algorithms to produce meaningful slices. In both cases, a key principle is to ensure that time shifts or cross-correlations do not go beyond the range of the currently inspected pulse. If this occurs, it can result in a mixing of results from multiple pulses or a reappearance of strong signals in later slices, leading to confusion in interpretation. As discussed previously, for up-chirp pulse trains, the later pulses tend to cluster more closely in time. This means that the narrowest pulse in such a pulse train is the second-to-last pulse, not the last pulse, as there is no pulse after it that would affect its results. Furthermore, since the quadrature signal is delayed relative to the in-phase signal, it is the second-to-last pulse of the quadrature signal that is closest to overlapping with the next pulse during time shifting. Hence, the available time range to produce slices is between the second-to-last quadrature pulse and the last in-phase pulse, as the last in-phase pulse coincides with the start of the response data of the last pulse. The number of available slices is then calculated as the result of dividing this window by the width of the slices, which can be expressed mathematically as:

$$[\text{Number of Available Slices}] = \frac{[\text{Length of Valid Slice Window}]}{[\text{Slice Width}]}$$

It should be further specified that for *eTCPCT*, it is the cross-correlation between the entire thermal data and the quadrature reference data that would lead to such overlap. As for *TCPCT*, it is the time delay of the quadrature reference data during truncation that would lead to this issue.

(3) Improve:

Two significant improvements have been implemented in this step.

Firstly, it has been observed that cross-correlation calculations, when performed using the central processing unit (*CPU*) of computers, take considerably more time compared to other computation steps. Additionally, in the *TCPCT* algorithm, cross-correlation is performed multiple times for each pixel, resulting in significant

computational overhead if the entire calculation is carried out in a single, simple for loop. To address this, Fourier transforms have been applied by past literature to the thermal response and reference data, allowing for faster cross-correlation calculations in the frequency domain as a simple multiplication [1]. Despite these optimizations, the processing time still remains lengthy, taking 12 hours to execute for a single set of parameters, as mentioned earlier. Therefore, further changes need to be made to reduce the processing time and improve efficiency.

To address this issue, the thesis writer enhanced the algorithm by leveraging the computational power of graphical processing units (*GPUs*) in computers to perform highly efficient parallel computing. This implementation is also with the assistance of PyTorch, a commonly used Python module for training neural networks [22]. Although PyTorch lacks a built-in function for direct cross-correlation calculations, it provides a similar operation known as convolution [23]. The relationship between the two operations is that the convolution of two signals, $x_1(t)$ and $x_2(t)$, is equivalent to the cross-correlation of the signals $x_1(t)$ and $x_2(-t)$. Thus, "conv1d", the command for one-dimensional convolution in PyTorch, can be used to implement cross-correlations. However, it is important to note that the laser excitation and thermal response data must be reshaped before being passed into the "conv1d" command to ensure successful execution.

The "conv1d" command performs the following transformation:

$$\begin{aligned} & [\text{Batch Size}] * [\text{Number of Input Channels}] * [\text{Length of the Signal Sequence}] \\ & \rightarrow [\text{Batch Size}] * [\text{Number of Output Channels}] * [\text{Length of the Output Sequence}] \end{aligned}$$

Specifically, the batch size indicates how many cross-correlations of the same kind are happening simultaneously, while the number of input channels indicates how many different kinds of cross-correlations are being performed. The third dimension represents the direction along which the cross-correlation occurs. In the case of *TCPCT*, where the same reference signal is cross-correlated with each thermal signal from all pixels, there is only one type of cross-correlation, and the number of input channels should be 1. Meanwhile, the batch size should be set to the total number of pixels in the entire frame. This requires reshaping the thermal response data from:

$$[\text{Full Frame Width}] * [\text{Full Frame Height}] * [\text{Total Frame Number}]$$

into:

$$([\text{Full Frame Width}] * [\text{Full Frame Height}]) * 1 * [\text{Total Frame Number}]$$

After the cross-correlation, the dimension of the output is:

$$([\text{Full Frame Width}] * [\text{Full Frame Height}]) * 1 * [\text{Length of Cross Correlation Results}]$$

Then, another reshaping step is necessary to convert the entire cross-correlation matrix back to its expected form, which is:

$$[\text{Full Frame Width}] * [\text{Full Frame Height}] * 1 * [\text{Length of Cross Correlation Results}]$$

Following the introduction of this improvement, performance statistics were gathered to demonstrate its significance. With the update successfully implemented, the time required per slice of calculation reduced to only 45 seconds, showcasing a substantial decrease in the time needed to produce 120 slices of results from 12 hours down to 1 hour and 30 minutes. This represents a remarkable speed increase of 8 times, highlighting the significant efficiency gain achieved through the implementation of the proposed improvement.

Furthermore, Google Colab, an online coding platform from Google, currently offers resources of advanced *GPU* graphics cards [24]. Although there is a cost associated with such usage, it supports simultaneous processing of several datasets through running multiple Python interfaces. The price for this service is \$1.5 *CAD* per hour per Python session [25].

Recently, a new research hypothesis has emerged in the context of *TCPCT*, suggesting that the peaks in the higher harmonics of the amplitude and phase results, i.e., the shorter peaks, may also contain important characteristics of the inspected defects. To enable further investigation in this direction, the thesis writer has also developed a function that facilitates the retrieval of results from non-highest peaks when utilizing *GPU* for computations. This function has been confirmed to be compatible with *GPU* usage and does not significantly impact efficiency.

The approach to implementing this identification method involves creating a binary mask with 0/1 values across the entire amplitude / phase result matrix, where the entries corresponding to the highest peaks for each pixel are set to zero. By multiplying this mask with the result matrix, the highest peaks can be removed. Subsequently, a second round of highest peak identification can be applied based on the result matrix after the peak removal, allowing for further analysis of the deeper peaks and their potential significance in characterizing the inspected defects.

➤ 5.8 Save the Amplitude and Phase Slices to Disk:

Finally, once the calculations of the amplitude and phase slices are complete, the results are saved as NumPy files and colormap images on the computer disk for future reference. As an additional contribution, the thesis writer implemented an image saving function that allows the slices to be saved with one number per pixel. This enables the slices to be easily imported into Paint software on a Windows computer system, where the pixel index can be identified, and graphs of amplitude or phase changes can be drawn against the slices to identify any abnormalities in the data. This feature provides a convenient way to visually analyze the amplitude and phase changes in the slices and identify any interesting features or anomalies.

6. Experimental Results:

This section presents the experimental results of *eTCPCT* on crack imaging, which comprises two stages. In the first stage (Section 6.1), the effects of varying experimental parameters on the *TCPCT* outputs are examined. Then, in the second stage (Section 6.2), an additional software called ImageJ is utilized to generate a comprehensive 3D imaging of the crack. It is important to note that the *TCPCT* results will be addressed in future work, as the accuracy of the improvements mentioned in the previous section is still under review.

➤ 6.1 Parameter Variation Testing:

eTCPCT measurements involve the adjustment of multiple parameters, such as pulse width, slice width, pulse number, pulse spacing, and camera frame rate. Notably, pulse spacing is determined by three additional parameters: starting chirp frequency, ending chirp frequency, and chirp duration. In the forthcoming discussions, these parameters will be systematically varied to assess their impact on the resulting amplitude and phase slices. It is anticipated that while there may be variations in image quality, modifying these parameters will not significantly affect the results, as all data processing is based on the same crack.

It is important to also highlight that the discussions will primarily focus on two types of analysis for comparing the results. The first analysis entails determining the slice number at which traces of the crack are no longer clearly visible on the phase images obtained from the calculations. Results from amplitude slices are not reported, as they are highly sensitive to varying emissivity and uneven heating, which can cause surface features to persist in these images longer than in phase images. This can lead to confusion when characterizing the subsurface feature of the crack. Figures 6.1 and 6.2 provide an example of such a comparison, where it is evident that the phase results exhibit a smoother trend across the slices compared to the amplitude results.

Meanwhile, the second analysis entails monitoring changes in the trajectory of the amplitude or phase values at a specific point on the sample surface while varying input parameters. Consistency in the results would be expected, with the trajectories following a similar trend.

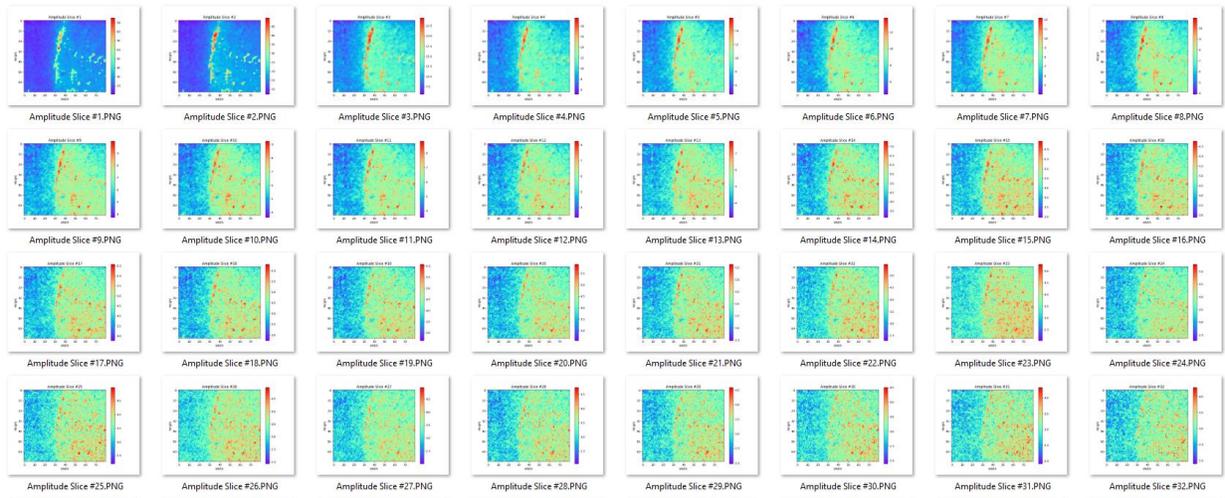


Figure 6.1. Sample Sequence of Amplitude Slices

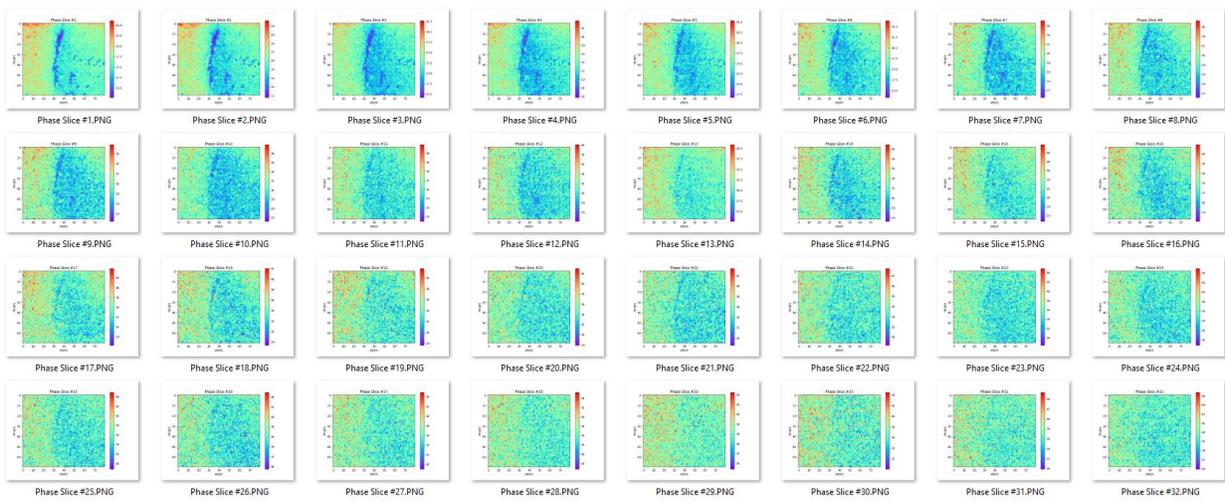


Figure 6.2. Sample Sequence of Phase Slices

The results from each type of parameter testing are provided in the following pages. To enhance the presentation in this report, comments and highlights from each test are compiled into a list as follows:

1. The phase channel consistently shows crack disappearance occurring between slices 22 and 29. Discrepancies may be attributed to imaging noise and human error in visually approximating the disappearance of the crack from the slices.
2. Varying pulse width significantly affects amplitude slices, with larger pulse widths resulting in higher magnitudes due to increased energy deposition on the surface. Phase discrepancies are comparatively smaller, but it can be observed that larger pulse widths also lead to smaller phase changes.
3. Varying slice width has a noticeable impact on amplitude results as it affects the pulse width of the reference signals used in the cross-correlations. Smaller slice widths can also help to smooth out phase variations, indicating that decreasing slice width can improve the axial resolution of the results.
4. Varying pulse number shows that considering more pulses decreases the phase increase over all the slices, which may be disadvantageous as a wider range of amplitude and phase results would lead to higher sensitivity in detection.
5. No apparent trends or effects are observed from the results of varying pulse spacing.
6. Lastly, varying camera frame rates shows that higher frame rates result in less variation in the data. This decrease in variation is not the same as in the case of pulse number, where decreased sensitivity would occur. Instead, it is helping to reduce noise in the signal. Therefore, higher frame rates should be used. It can be observed from the figures that the phase variation due to noise for cases of 100 Hz, 200 Hz, and 400 Hz are approximately 40°, 20°, and 10°, respectively. Hence, this confirms the well-known signal processing conclusion that increasing sampling rate improves the signal-to-noise ratio of the output.

(1) Varying pulse width:

Table 6.1. Slice Indices or Delay Time for Crack Disappearance from the Phase Images

Cases	Results
(1) Varying Pulse Width	
Pulse Width = 100 ms	Slice Index = 24
Pulse Width = 50 ms	Slice Index = 25
Pulse Width = 25 ms	Slice Index = 27
Pulse Width = 10 ms	Slice Index = 22

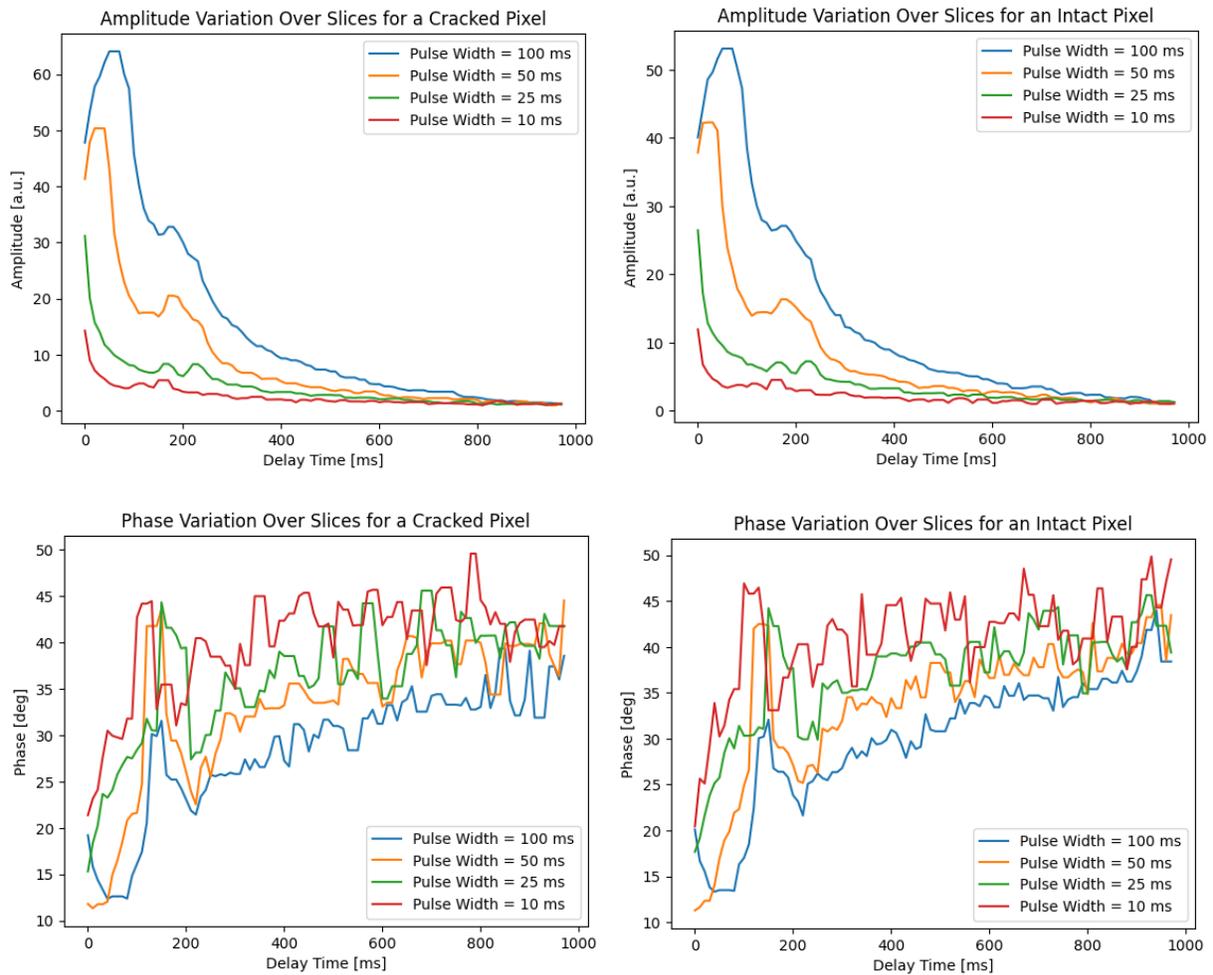


Figure 6.3. Variations of Amplitude and Phase Results for Cracked and Intact Pixels Under Different Pulse Widths

(2) Varying slice width:

Table 6.1. Slice Indices or Delay Time for Crack Disappearance from the Phase Images

(2) Varying Slice Width	
Cases	Results
Slice Width = 100 ms	Delay Time = 200 ms
Slice Width = 50 ms	Delay Time = 200 ms
Slice Width = 20 ms	Delay Time = 240 ms
Slice Width = 10 ms	Delay Time = 240 ms

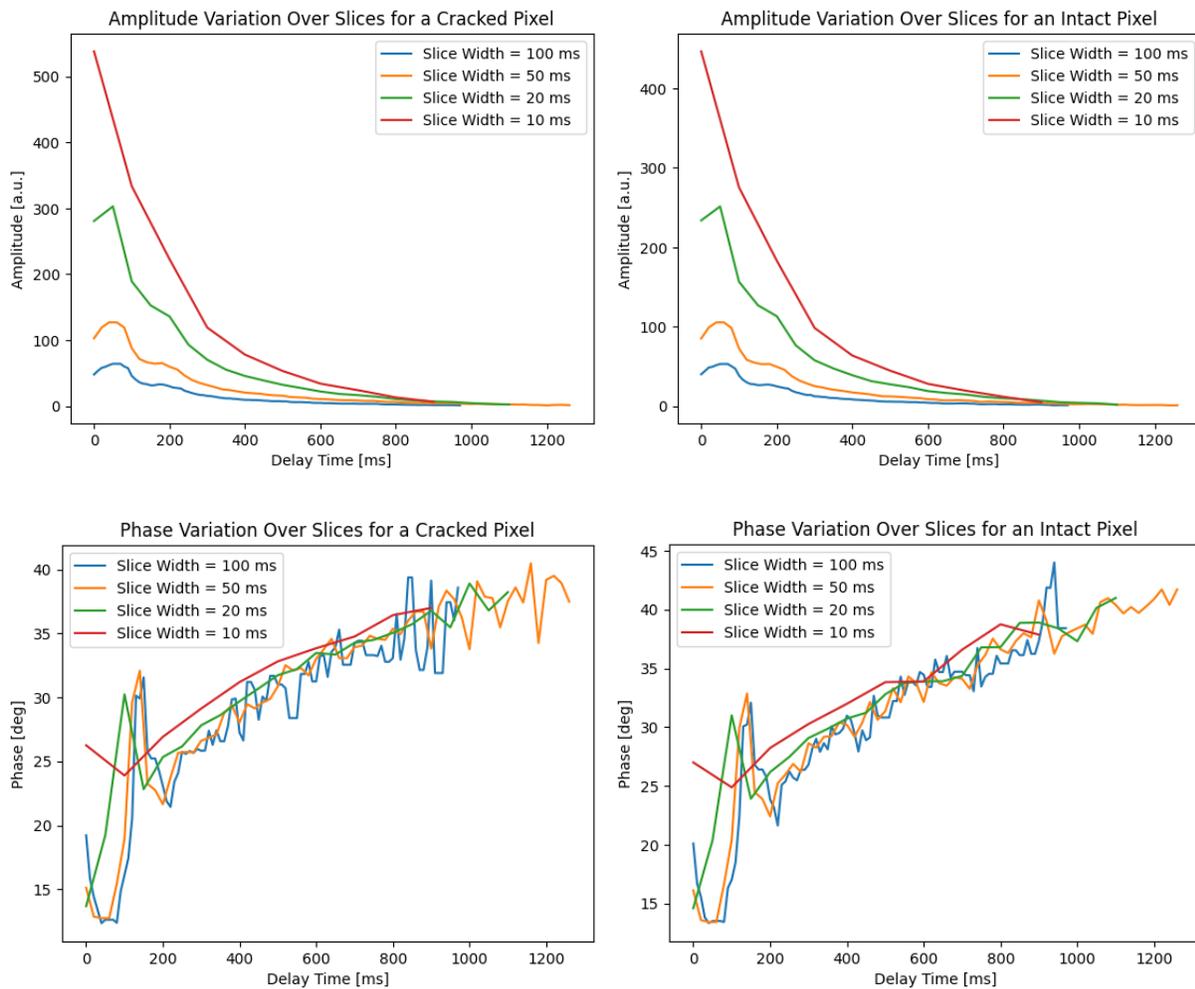


Figure 6.4. Variations of Amplitude and Phase Results for Cracked and Intact Pixels Under Different Slice Widths

(3) Varying pulse number:

Table 6.1. Slice Indices or Delay Time for Crack Disappearance from the Phase Images

(3) Varying Pulse Number	
Cases	Results
Pulse Number = 9	Slice Index = 29
Pulse Number = 5	Slice Index = 28
Pulse Number = 2	Slice Index = 23

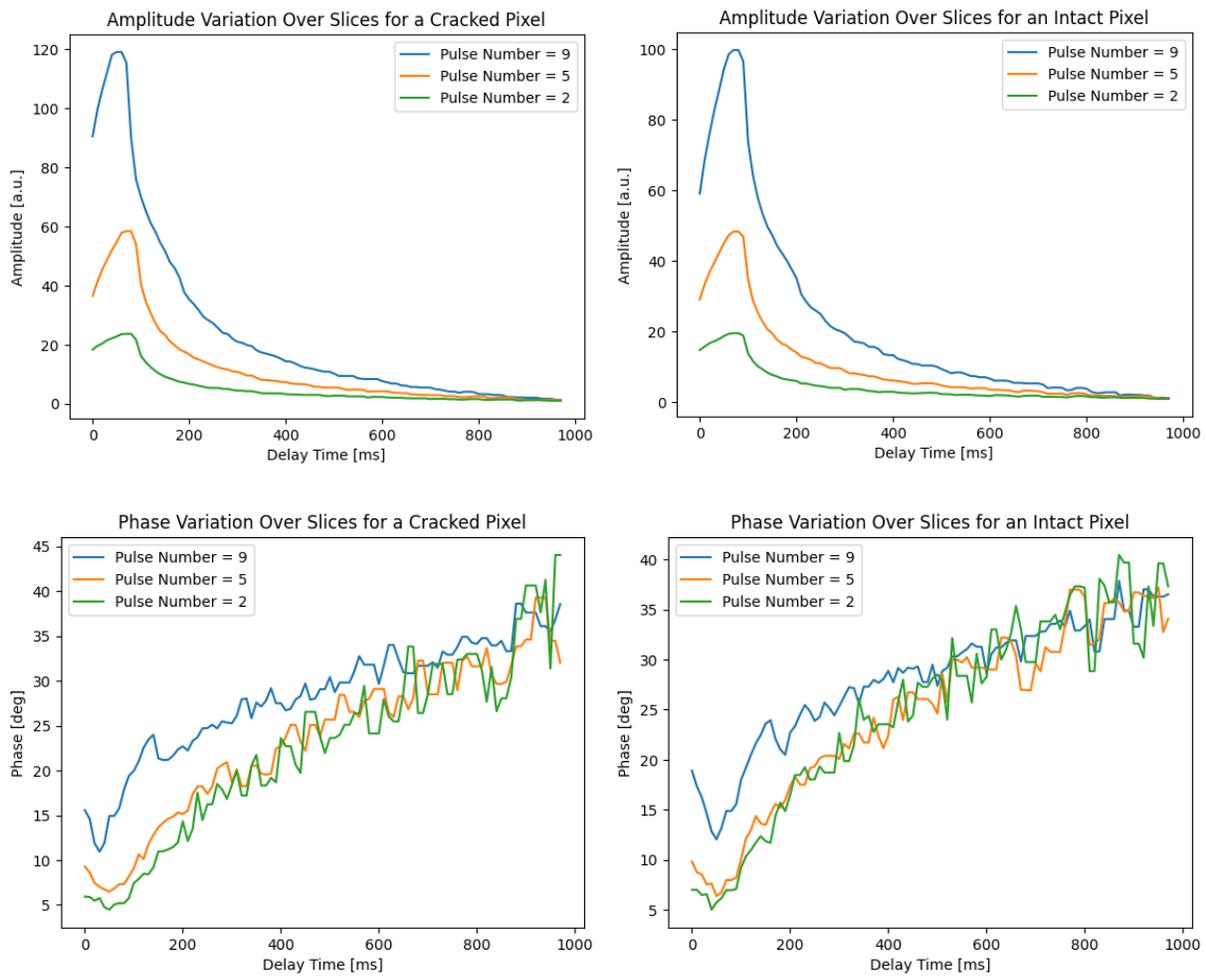


Figure 6.5. Variations of Amplitude and Phase Results for Cracked and Intact Pixels Under Different Pulse Numbers

(4) Varying pulse spacing:

Table 6.14 Slice Indices or Delay Time for Crack Disappearance from the Phase Images

(4) Varying Pulse Spacing (All First 5 Pulses Selected)	
Cases	Results
Excitation Chirp = 0.2~0.6 Hz, 14 s, 6 pulses	Slice Index = 26
Excitation Chirp = 0.2~0.6 Hz, 24 s, 10 pulses	Slice Index = 21
Excitation Chirp = 0.2~1.2 Hz, 14 s, 10 pulses	Slice Index = 27
Excitation Chirp = 1~1.2 Hz, 16 pulses	Slice Index = 28

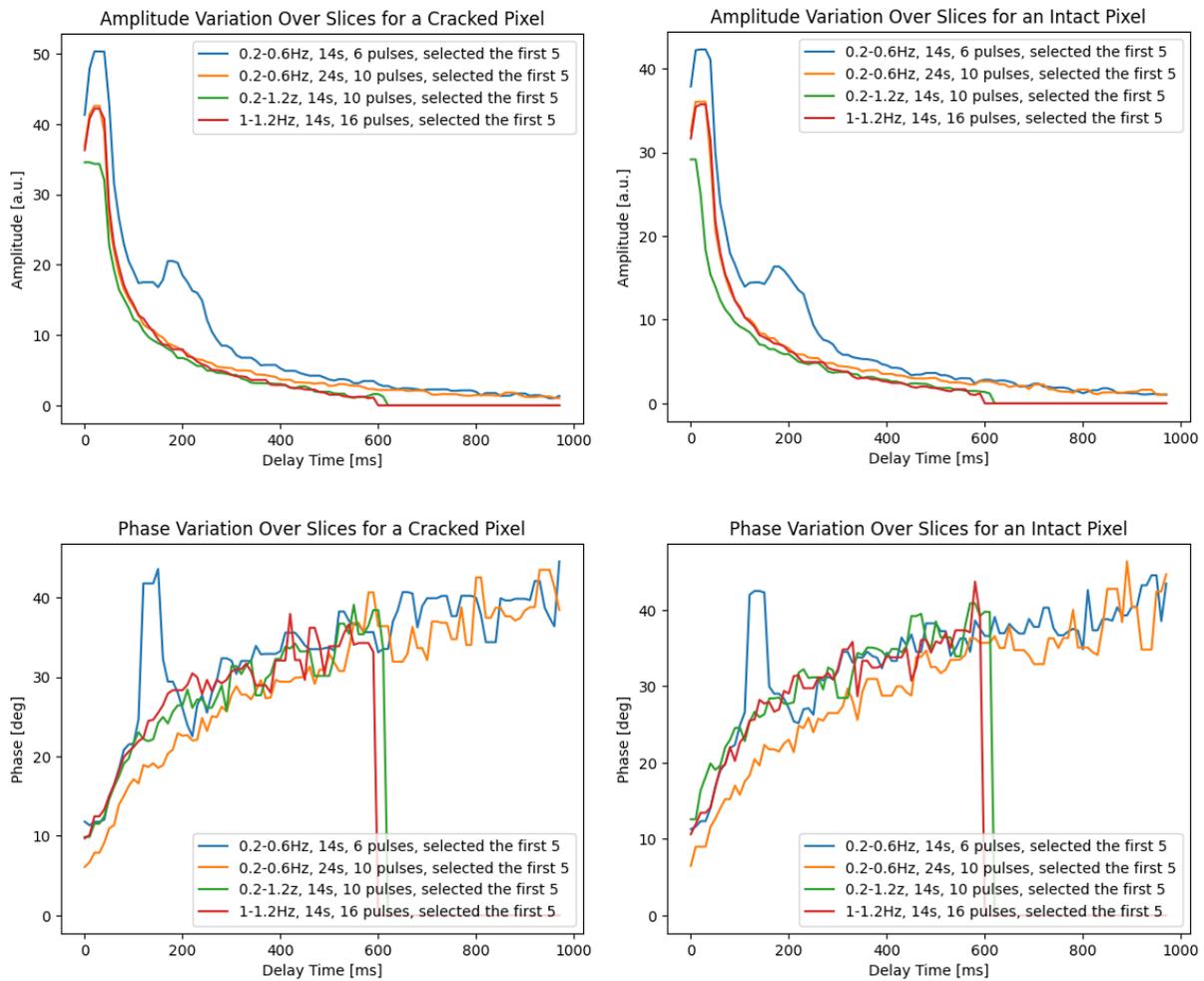


Figure 6.6. Variations of Amplitude and Phase Results for Cracked and Intact Pixels Under Different Pulse Spacings

(5) Varying camera frame rate:

Table 6.5. Slice Indices or Delay Time for Crack Disappearance from the Phase Images

(5) Varying Camera Frame Rate	
Cases	Results
Camera Frame Rate = 100 Hz	Slice Index = 28
Camera Frame Rate = 200 Hz	Slice Index = 26
Camera Frame Rate = 400 Hz	Slice Index = 28

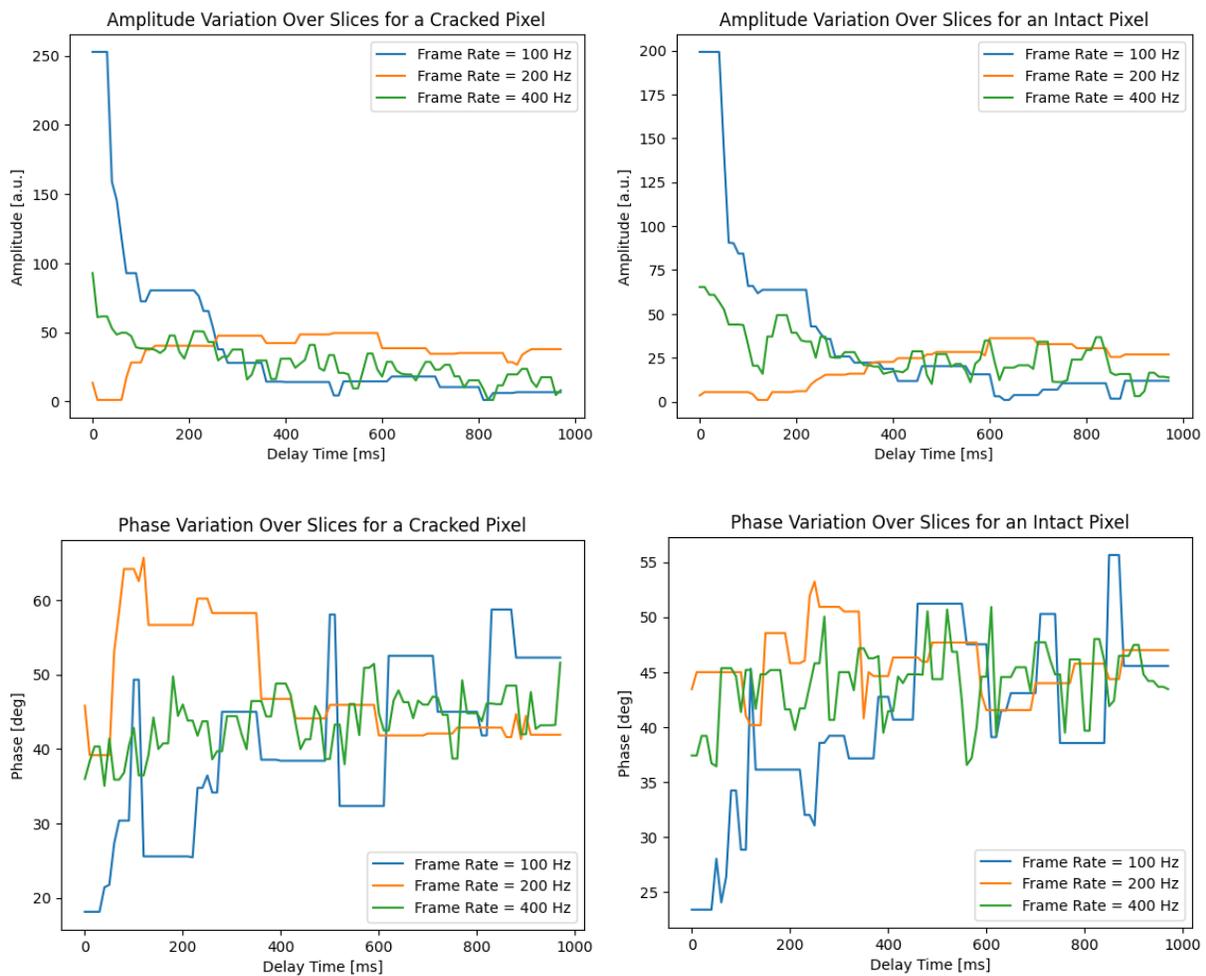


Figure 6.7. Variations of Amplitude and Phase Results for Cracked and Intact Pixels Under Different Frame Rates

➤ **6.2 3D Crack Imaging:**

This last section of the report presents the ongoing work on 3D crack imaging. Although this step has not been applied to all datasets yet, some promising results have already emerged, as evidenced by the figures below. It is important to note that an image software called ImageJ has been utilized to enable such visualizations. This software enables zooming in on a specific section of the calculated slices, as depicted in Figure 6.8, and displaying only the cross section of that particular section. An example stack displayed in ImageJ's 3D visualizer is shown in Figure 6.9. It is evident from the figure that the crack extends 1/4 into the thickness of the entire stack, affirming that the TCPCT algorithm indeed possesses 3D crack profiling capabilities.

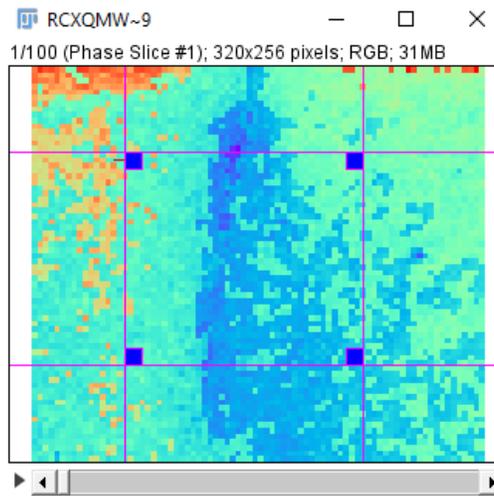


Figure 6.8. Screenshot Demonstrating ImageJ's Stack Slicing Ability

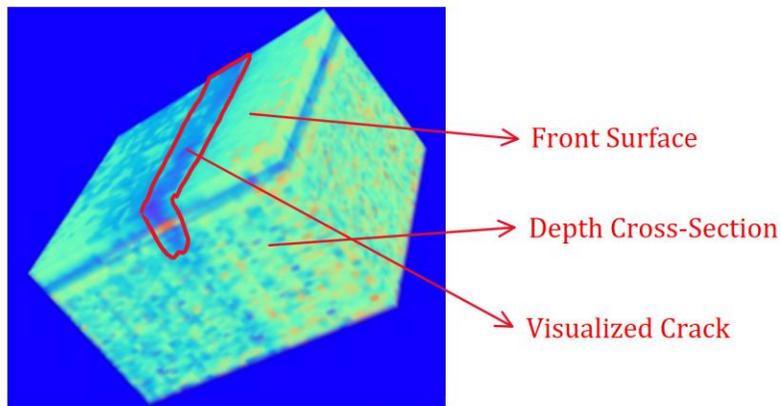


Figure 6.9. Sample 3D Crack Visualization Results

7. Summary:

To conclude the thesis, this work provides a comprehensive review and introduces significant advancements to the *TCPCT* algorithm. The methodology's history, setup, and algorithm are thoroughly examined for review purposes. Additionally, novel improvements to the algorithm's execution are proposed, along with its application to crack imaging, a previously unexplored topic. The following list highlights the latest advancements in *TCPCT* presented in this work:

1. A more robust peak search method is introduced for analyzing pulse thermography data. This method utilizes peak prominence and peak width to accelerate the process and to enhance identification accuracy.
2. A more efficient cross-correlation computation method is developed using Python's PyTorch module, enabling parallel computation through convolution commands. This improvement results in an 8-fold increase in computation speed.
3. Controlled experiments are conducted to evaluate the sensitivity of *TCPCT* to different input parameter variations.
4. 3D visualization of cracks is achieved by passing amplitude and phase stacks into the ImageJ software. This allows for obtaining cross-sectional views of cracks in the depth direction, enabling identification of crack depth and other details.

Overall, this thesis contributes to the advancement of *TCPCT* by proposing innovative methods for peak search, computation, sensitivity analysis, and crack visualization, enhancing the algorithm's performance and expanding its application to crack imaging.

8. References:

- [1] Sreekumar Kaiplavil and Andreas Mandelis, "Truncated-Correlation Photothermal Coherence Tomography for Deep Subsurface Analysis," *Nature Photonics*, pp. 635–642, Jun. 2014, doi: 10.1038/nphoton.2014.111.
- [2] Sreekumar Kaiplavil, Andreas Mandelis, Xueding Wang, and Ting Feng, "Photothermal Tomography for the Functional and Structural Evaluation, and Early Mineral Loss Monitoring in Bones," *Biomedical Optics Express*, vol. 5, no. 8, p. 2488, 2014, doi: 10.1364/BOE.5.002488.
- [3] Sreekumar Kaiplavil, Andreas Mandelis, and Bennett T. Amaechi, "Truncated-Correlation Photothermal Coherence Tomography of Artificially Demineralized Animal Bones: Two- and Three-Dimensional Markers for Mineral Loss Monitoring," *Journal of Biomedical Optics*, vol. 19, no. 2, 2014, doi: 10.1117/1.JBO.19.2.026015.
- [4] Sohrab Roointan, Pantea Tavakolian, Koneswaran S. Sivagurunathan, Marie Floryan, Andreas Mandelis, and Stephen H. Abrams, "3D Dental Subsurface Imaging Using Enhanced Truncated Correlation-Photothermal Coherence Tomography," *Scientific Reports*, vol. 9, no. 1, p. 16788, Dec. 2019, doi: 10.1038/s41598-019-53170-w.
- [5] Sohrab Roointan, Pantea Tavakolian, Koneswaran S. Sivagurunathan, Andreas Mandelis, and Stephen H. Abrams, "Detection and Monitoring of Early Dental Caries and Erosion Using Three-Dimensional Enhanced Truncated-Correlation Photothermal Coherence Tomography Imaging," *Journal of Biomedical Optics*, vol. 26, no. 4, Apr. 2021, doi: 10.1117/1.JBO.26.4.046004.
- [6] Elnaz Baradaran Shokouhi, Robert Welch, Koneswaran Sivagurunathan, and Andreas Mandelis, "Multispectral Truncated-Correlation Photothermal Coherence Tomography Imaging Modality for Detection of Early Stage Dental Caries," *Biomedical Optics Express*, vol. 13, no. 5, pp. 2772–2781, May 2022, doi: 10.1364/BOE.455117.
- [7] Robert Welch, Koneswaran Sivagurunathan, Pantea Tavakolian, and Andreas Mandelis, "Three-Dimensional Thermophotonic Image Optimization Modalities of Truncated Correlation Photothermal Coherence Tomography," *Journal of Biophotonics*, vol. 15, no. 7, 2022, doi: 10.1002/jbio.202200018.
- [8] D. Thapa *et al.*, "Comparison of Long-Wave and Mid-Wave Infrared Imaging Modalities for Photothermal Coherence Tomography of Human Teeth," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 9, pp. 2755–2766, 2022, doi: 10.1109/TBME.2022.3153209.
- [9] P. Tavakolian, S. Sfarra, G. Gargiulo, K. Sivagurunathan, and A. Mandelis, "Photothermal Coherence Tomography for 3-D Visualization and Structural Non-Destructive Imaging of a Wood Inlay," *Infrared Physics & Technology*, vol. 91, pp. 206–213, 2018, doi: 10.1016/j.infrared.2018.04.018.
- [10] Pantea Tavakolian, Elnaz B. Shokouhi, Stefano Sfarra, Gianfranco Gargiulo, and Andreas Mandelis, "Non-Destructive Imaging of Ancient Marquetries Using Active Thermography and Photothermal Coherence Tomography," *Journal of Cultural Heritage*, vol. 46, pp. 159–164, Nov. 2020, doi: 10.1016/j.culher.2020.05.002.

- [11] Pantea Tavakolian, Koneswaran Sivagurunathan, and Andreas Mandelis, "Enhanced Truncated-Correlation Photothermal Coherence Tomography with Application to Deep Subsurface Defect Imaging and 3-Dimensional Reconstructions," *Journal of Applied Physics*, vol. 122, no. 2, 2017, doi: 10.1063/1.4992807.
- [12] P. Tavakolian and A. Mandelis, "Perspective: Principles and Specifications of Photothermal Imaging Methodologies and Their Applications to Non-Invasive Biomedical and Non-Destructive Materials Imaging," *Journal of Applied Physics*, vol. 124, no. 16, 2018, doi: 10.1063/1.5044748.
- [13] Hai Zhang, Pantea Tavakolian, Koneswaran Sivagurunathan, Andreas Mandelis, Wei Shi, and Fei-Fei Liu, "Truncated-Correlation Photothermal Coherence Tomography Derivative Imaging Modality for Small Animal in Vivo Early Tumor Detection," *Optics Letters*, vol. 44, no. 3, p. 675, 2019, doi: 10.1364/OL.44.000675.
- [14] Pantea Tavakolian, Sohrab Roointan, and Andreas Mandelis, "Non-Invasive in-Vivo 3-D Imaging of Small Animals Using Spatially Filtered Enhanced Truncated-Correlation Photothermal Coherence Tomography," *Scientific Reports*, vol. 10, no. 1, 2020, doi: 10.1038/s41598-020-70815-3.
- [15] Ali Risheh, Pantea Tavakolian, Alexander Melnikov, and Andreas Mandelis, "Infrared Computer Vision in Non-Destructive Imaging: Sharp Delineation of Subsurface Defect Boundaries in Enhanced Truncated Correlation Photothermal Coherence Tomography Images Using K-Means Clustering," *NDT & E International*, vol. 125, 2022, doi: 10.1016/j.ndteint.2021.102568.
- [16] "Fixed Focus Collimation Packages: SMA905 Connectors." <https://www.thorlabs.com> (accessed Apr. 16, 2023).
- [17] Thorlabs, " $\lambda=633\text{nm}$, SMA Collimation Package, $\text{NA}=0.25$, $f=10.99\text{mm}$."
- [18] "Chirp," *Wikipedia*. Feb. 26, 2023. Accessed: Apr. 15, 2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Chirp&oldid=1141754848>
- [19] T. Tulka, "LoRa Spreading Factor Explained by Tomas Tulka." <https://blog.tulka.com/lora-spreading-factor-explained> (accessed Apr. 15, 2023).
- [20] "Prominence - MATLAB & Simulink." <https://www.mathworks.com/help/signal/ug/prominence.html> (accessed Apr. 17, 2023).
- [21] Basj, "Answer to 'Peak-finding algorithm for Python/SciPy,'" *Stack Overflow*, Oct. 02, 2018. <https://stackoverflow.com/a/52612432> (accessed Apr. 17, 2023).
- [22] "PyTorch." <https://www.pytorch.org> (accessed Apr. 18, 2023).
- [23] "Conv1d — PyTorch 2.0 documentation." <https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html> (accessed Apr. 18, 2023).
- [24] "Google Colab." <https://research.google.com/colaboratory/faq.html> (accessed Apr. 18, 2023).
- [25] "Google Colab." <https://colab.research.google.com/signup> (accessed Apr. 18, 2023).