A PHOTO-THERMAL NON-DESTRUCTIVE EVALUATION METHOD FOR CERAMICS WITH APPLICATIONS FOR OTHER MATERIALS

BY

MARCUS JEANFREAU LA GRONE

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE

May, 1995

A PHOTO-THERMAL NON-DESTRUCTIVE EVALUATION METHOD FOR CERAMICS WITH APPLICATIONS FOR OTHER MATERIALS

Thesis Approved:

Thesis Advisor

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincere appreciation for the help, time, patience, knowledge, and guidance of my advisor, Dr. Jerzy Krasinski. His assistance and effort was paramount for my success in lab. More than just a lab mentor, he has taught me more in two years outside the lab than I learned in all of public education. I would also like to extend my thanks to Dr. Czeslaw Radzewicz for his dedication, knowledge, patience, and his humor. I would also like to thank all those in the Department of Electrical and Computer Engineering, both faculty and students, that helped me with this project.

I would also like to thank the late Allen Melton, a teacher from high-school that encouraged me to explore the world around me, and to set standards for academia and life that would be worthy of his example.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF EQUATIONS

CHAPTER 1.

INTRODUCTION

Ceramics have made themselves a home in the modern world of high-technology. Being very strong and able to withstand high temperatures and harsh environments, ceramics are well suited for applications ranging from high-speed turbines, to ball bearings, to both thermal and ballistic shielding. Ceramics used in these high-stress systems all share one major problem: catastrophic failure due to undetected growth of cracks. Due to their very nature and construction, many ceramics resist sub-surface inspection from traditional test methods such as X-rays and ultrasonic because of the similar size of the defects and the crystals [1]. Many nondestructive evaluation (NDE) methods have been proposed to deal with the problem of subsurface detection in ceramics. It is the intention of this report to discuss a Photo-Thermal method of inspection that is suitable for the detection of cracks and subsurface defects in ceramics as well as other materials.

Photo-Acoustic methods have previously been explored as a tool for defect detection. One such method used a modulated CW laser source to probe a ceramic, and detected the recoil of the sample as the heated part of the sample expanded [2]. Such a method relies on the fact that the thermal diffusion through the material is radically modified by boundary layers (i.e. cracks). Since the heat does not quickly diffuse across the crack boundary, the areas near the crack will hold more heat and thus expand more. The Photo-Thermal system relies on this same phenomenon, yet instead of trying to detect the recoil of the sample, the system directly looks for the infrared (IR) energy radiated from the sample[3].

Other people have experimented with PT inspection systems. Indeed systems using a computer controlled strobe and a highly sensitive, wide spectrum IR camera have demonstrated a system capable of detecting some sub-surface defects [4][5]. However, due to their choice of wide-spectrum detection optics they failed to capitalize on a particular feature of black body emissions.

CHAPTER II

METHODOLOGY

THERMAL DIFFUSION

When the laser strikes the surface of the ceramic, part of the energy is absorbed and goes to heating the material. If one considers a laser beam with a typical gausian profile, the description of the heating of the surface becomes difficult; the mathematical description of the heat diffusing through the material becomes quite complex. If one, treats the laser as a point source, the heat diffusion can be modeled as a spherical shell expanding outwards from the source. How far this heat wave can propagate is dependent on the material. [6] For most materials the thermal diffusion can be described as:

$$\alpha = k / C \rho \qquad (1)$$

Where k is the thermal conductivity, C is the specific heat, and ρ is the specific gravity of the material. As the laser is being chopped, it is useful to describe the thermal diffusion in terms of this chopping frequency:

$$\mu = (2 \alpha / \omega)^{1/2} \qquad (2)$$

Where μ is the thermal diffusion length. It is quickly apparent that the frequency is critical in determining the depth of these "heat waves."

In a uniform material, these heat waves would propagate evenly and heat the material uniformly. Any inhomoginiety in the material (cracks or change material properties) would result in variations in the penetration of these heat waves. A shallow penetration would mean more of the energy is trapped nearer the surface, hence the surface would be hotter. By detecting these subtle variations in the heating of the surface, one should be able to detect subsurface imperfections.

BLACKBODY RADIATION

Black body radiation is now a well understood effect. In 1900 Max Planck derived an equation that accurately models the black body radiation [7].

$$f(\lambda) = \frac{8\,\pi\,h\,c\,\lambda^{-5}}{e^{hc/\lambda kT}-1} \qquad (3)$$

Examples of the output spectrum are shown on Figure 1. for 100-500 degrees Celsius. It can be noted that as the temperature rises, the peak not only grows but also shifts to shorter wavelengths. The total energy radiated by the system grows as a function of temperature to the fourth power.

$$I=\sigma\,T^4 \qquad (4)$$

As such, small changes in temperature can account for noticeable changes in the total power of the system.

This dependence, QE.(4), holds only if one is to observe the entire spectrum. In order to gain as much energy as possible, wide spectrum detectors are commonly used. If a narrow "window" of the spectrum is used one will obviously not collect as much energy. However, since the peak is also sliding towards shorted wavelengths as temperature increase, the *change* in energy as a function of temperature can be much greater than the simple fourth power dependence. Such a window would need to sit such that it catches the encroaching edge of the curve. In this way the apparent dependency on temperature be effected not only by the area under the curve, but also by the fact that the curve is sliding deeper into the window

Figure 2. shows the results of energy dependence as a function of window width. The window starts at .3 microns and then ranges from 1 to 10 microns. In the case that the window get larger, the dependence approaches the expected value of "4". For narrow windows this dependence is much greater.

At this point their is a trade-off, higher sensitivity to temperature change versus total signal. Cryogenically cooled detectors offer very low noise and lock-in amplifiers offer the ability to isolate a pulse synchronized with the reference from the rest of the environment. Thus is possible to get a reproducible signal and at the same time get a dependency of better than temperature to the fourteenth power.

Most photo-thermal systems use a pulsed heat source. This permits precise synchronization with the test equipment and, more importantly, permits one to vary the frequency of the pulses. The rate of the pulse has a direct influence on the apparent "depth" that the system can sense. The slower the repetition rate the more power is dumped to the system per unit pulse. Slower repetition rates also give the material time to diffuse the heat deeper into the system. The trade-off is that this diffusion spreads in a spherical wave front. Thus the deeper it penetrates the wider the area of influence, thus less resolution. By using a laser as the pulsed heat source one can focus the beam to help change resolution.

Hence a variably modulated laser source that was synchronized to a lock-in amplifier was used in conjunction with a narrow window infra-red detector. This offered the possibility of a high level of distinction and a moderate signal strength.

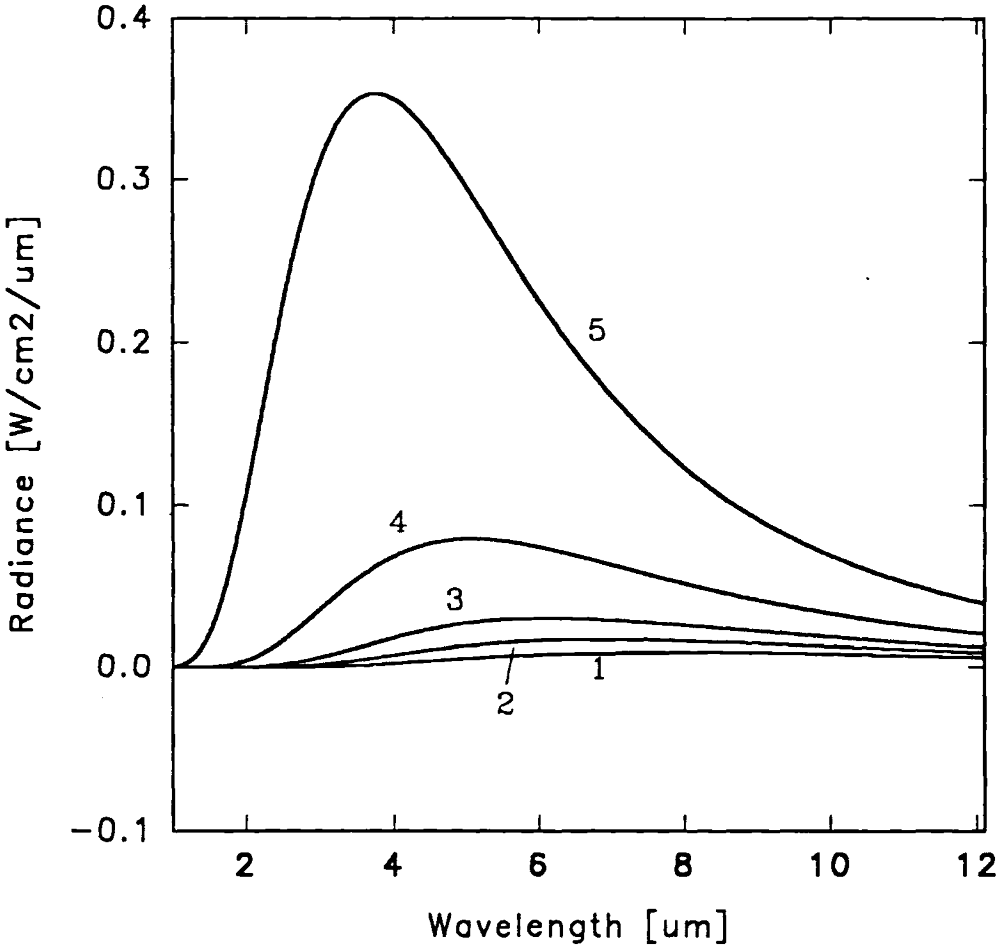


Figure 1.

Blackbody radiation curve for various temperatures: 1) 100 °C 2) 150 °C, 3) 200 °C, 4) 300 °C, 5) 500 °C. As the temperature rises, the peak of the curve rises and shifts toward shorter wavelengths.

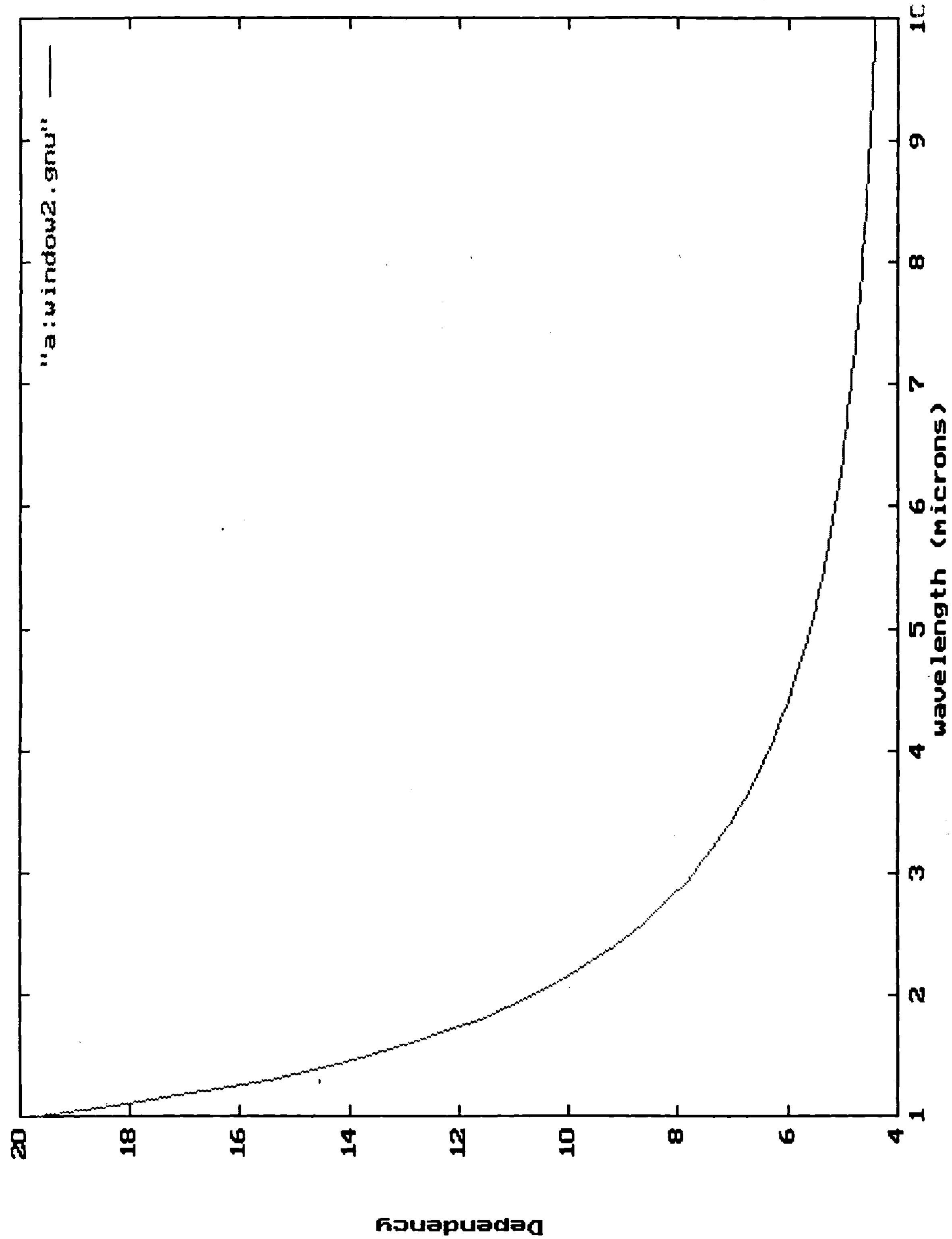
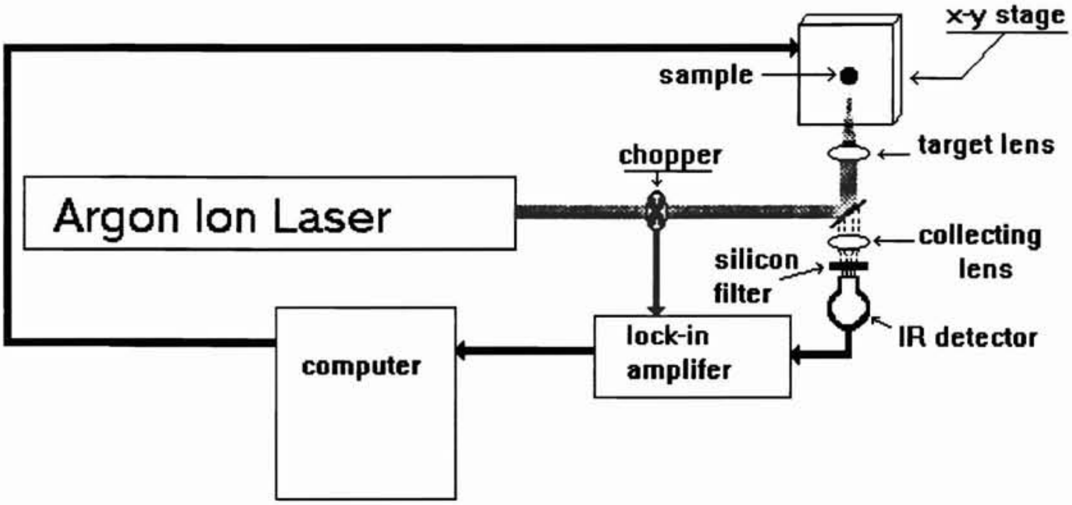


Figure 2.

Dependency Vs low pass window width. For small windows there is a dramatic non-linearity. Larger windows more closely correspond to the expected fourth order dependency.

CHAPTER III

EXPERIMENTAL SYSTEM



The light from the argon laser is modulated at a fixed frequency by the chopper that is linked to the lock-in amplifier. This modulated light is then reflected off of a mirror that is highly reflective to green light, yet transparent to infrared. The light is then focused onto the sample. The laser energy heats a small area of the sample and emits infrared. This infrared radiation then passes back through the optics, past the mirror, is filtered to remove any traces of visible light, and is detected by the infrared detector.

The signal from the detector and the synchronization pulse from the chopper are feed into the lock-in amplifier. This style amplifier has the advantage of being able to detect a signal much weaker than the noise. The clean signal is then read and stored by the master computer. While this is being done, the computer moves the sample using via the x-y stage. This permits the sample to be scanned in a controlled and repeatable fashion.

Details of the System

Laser

A Spectra Physics "171" Argon Ion laser was used. This is an industry standard water cooled, gas-discharge laser that can operate between 450 and 514nm and can reach powers in excess of 15 watts of continuous optical power. For the experiments, the laser was operated at 514nm and powers of 20-150mW. Powers of 4-6 W were used in order to mark ceramics for control studies.

Lens

The target lens was a BK-7 glass. Focal lengths from 10mm out to 75mm were used in order to gain a variety of spot sizes and intensities. The collection lens was usually a BK-7 glass with a focal length of 25mm. Later experiments used a Zinc-Selenide lens with a focal length of 25mm for the collection lens. ZnSe lens offer superior optical properties at farther IR wavelengths.

Filter

A silicon window was used to filter out unwanted light. Such a window will permit 55% of the 1-8 micron light past, but will let less than 1% of visible light pass through.

Lock-In Amplifier

A Standford Research "SR-510" lock-in amplifier was used. The model can handle signal strengths from the nanovolt range up to millivolts. It has a wide selection of options in terms of controlling the phase angle, pre- and post- time constants, as well as having a easy to use computer interface. However, for gathering data at a high speed it was necessary to use a faster sampler than the one in the SR-510. As the SR-510 offers a conditioned signal out; this was used for the higher speed scans in order to reduce the noise.

Sampling Card

A National Instruments "Lab-PC" card was used to handle the high speed sampling. The card offers 12-bit resolution at rates of up to 62kHz. As it has adjustable bias and sampling range, it was easy to interface with the output of the SR-510. The card offers a linear response of better than .5%. This assures that the data from the SR-510 was gathered as accurately as possible.

Infrared Detector

A detector manufactured by Hamamatsu was used. When cooled with liquid nitrogen it offered a very good sensitivity as well as low noise levels. Its germanium sensor was suitably centered in the near-infrared (1.5 microns).

X-Y stage

The computer controlled translation stage was manufactured by Physik Instrumente. It offered sub-micron precision over a range of approximately one inch square. Its simple computer control made it easy to incorporate into the system.

Computer

A 80486-33 was used to run the system. As most of the control boards for lab equipment are designed for PCs, this was a reasonable choice. Its speed, coupled with the sampling board, permitted a high sampling frequency. This permitted extensive averaging the samples, further reducing the noise of the system.

CHAPTER IV

RESULTS

STATIONARY TEST

The first test of the system was one to determine if the non-linearity described by the math would hold for a real system. Figure 4 shows the results of sweeping the laser power versus the output signal. The system starts off linear, this is consistent with the target fluorescing at IR wavelengths. As the power increase, the resulting signal becomes progressively non-linear. With the silicon-nitride sample tested, it was possible to go to 200mW without damaging the material. At such levels, the measured non-linearity is on the order of the ninth power.

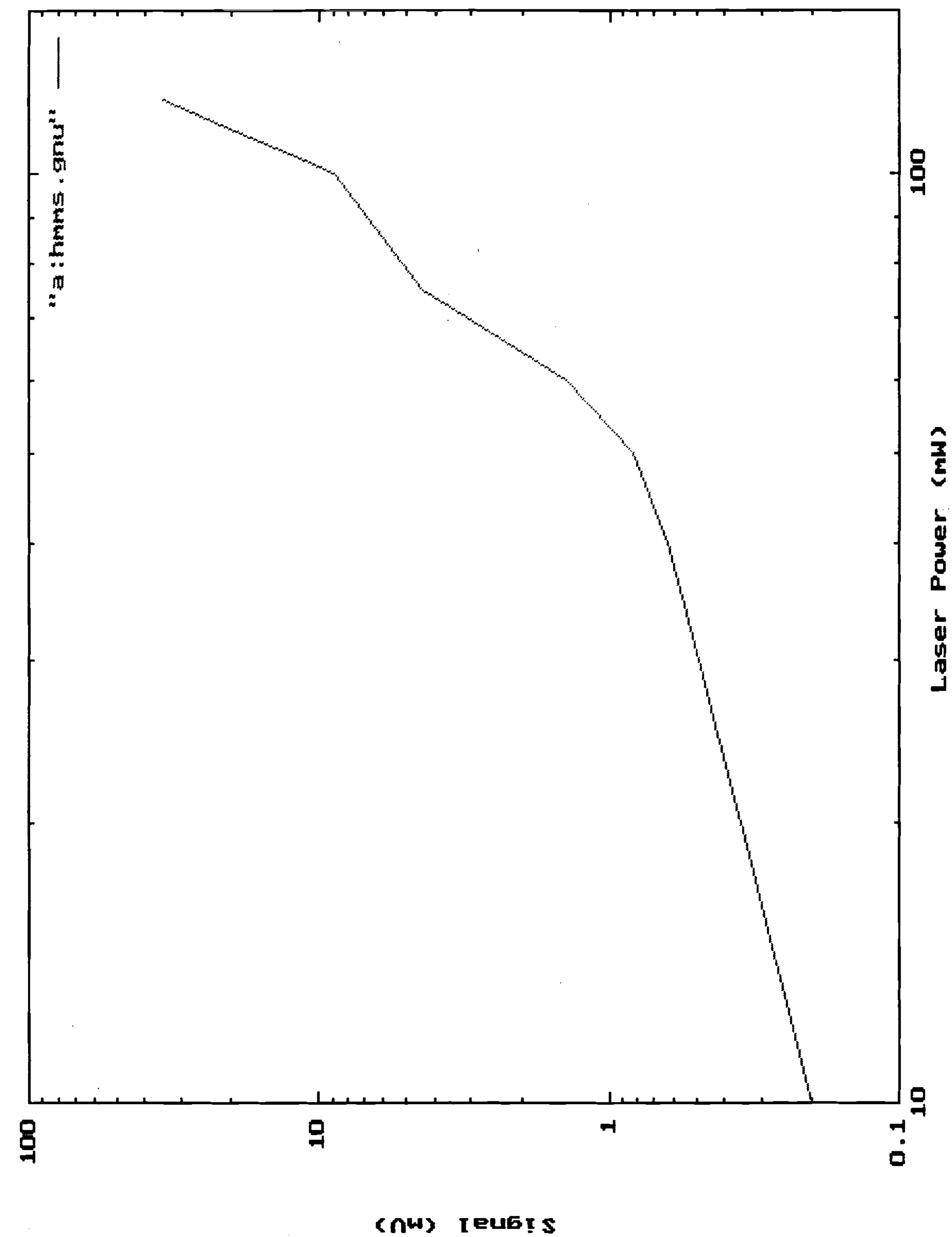


Figure 4.

Stationary test of nonlienarity. At low power fluorescence dominates the response. At higher power, a ninth order non-linearity dominates the response.

CONTROL CASE: UNDERCUT CERAMIC

In order to test the system it was decided to create a sample that had numerous flaws and a controlled thickness. The controlled thickness was important to give some idea as to the effective penetration depth in this ceramic. The sample was prepared by cutting a channel on the under side by using a diamond saw. Figures 5 and 6 show a wafer of ceramic that was undercut with a diamond saw (figure 7 shows sample orientation). Figure 5 is a tightly focused beam showing great surface and sub-surface details. Figure 6 shows approximately the same area by using a wide beam and a slower frequency. The wider beam removes much of the minute details while the slower frequency permits a deeper penetration of the heat waves into the sample. In this case, a lower resolution was chosen in order to average out the signal, permitting one to compare it with the expected results of a controlled thickness (compare Figure 6 with the thickness model in Figure 7.) Figure 8. shows a photograph of the area taken with an optical microscope.

While the signal did show a good correlation with the thickness of the material, the excessive number of cracks caused by the saw diminished the ability to determine the precise penetration depth of the probe beam. However, it did quite clearly show many cracks inside the material that were not apparent from the surface.

250 microns

800 microns

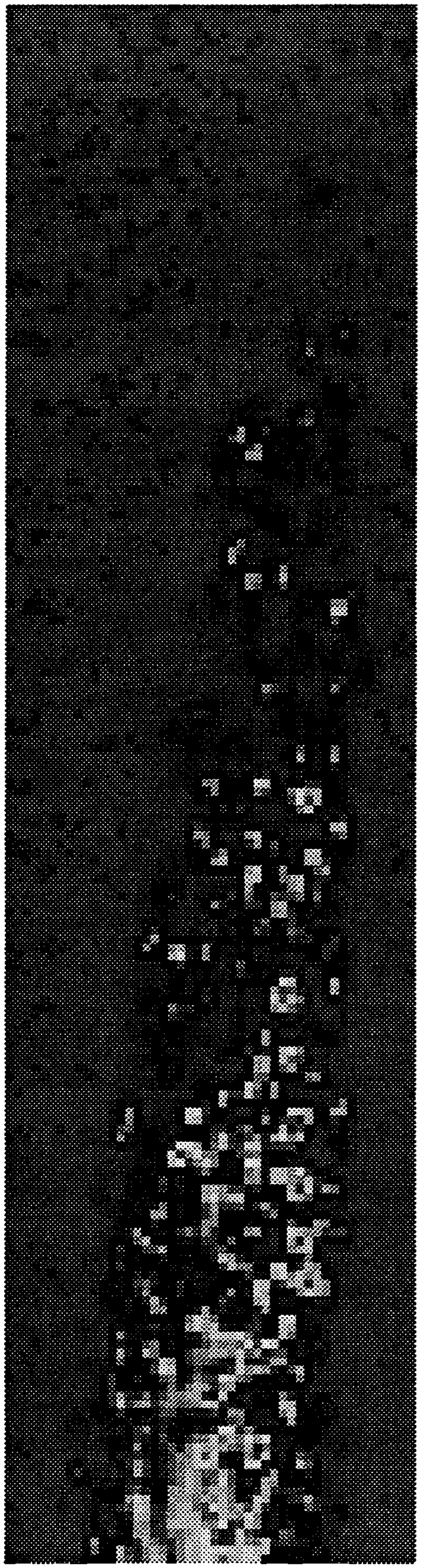


Figure 5.

High-detail of undercut ceramic.

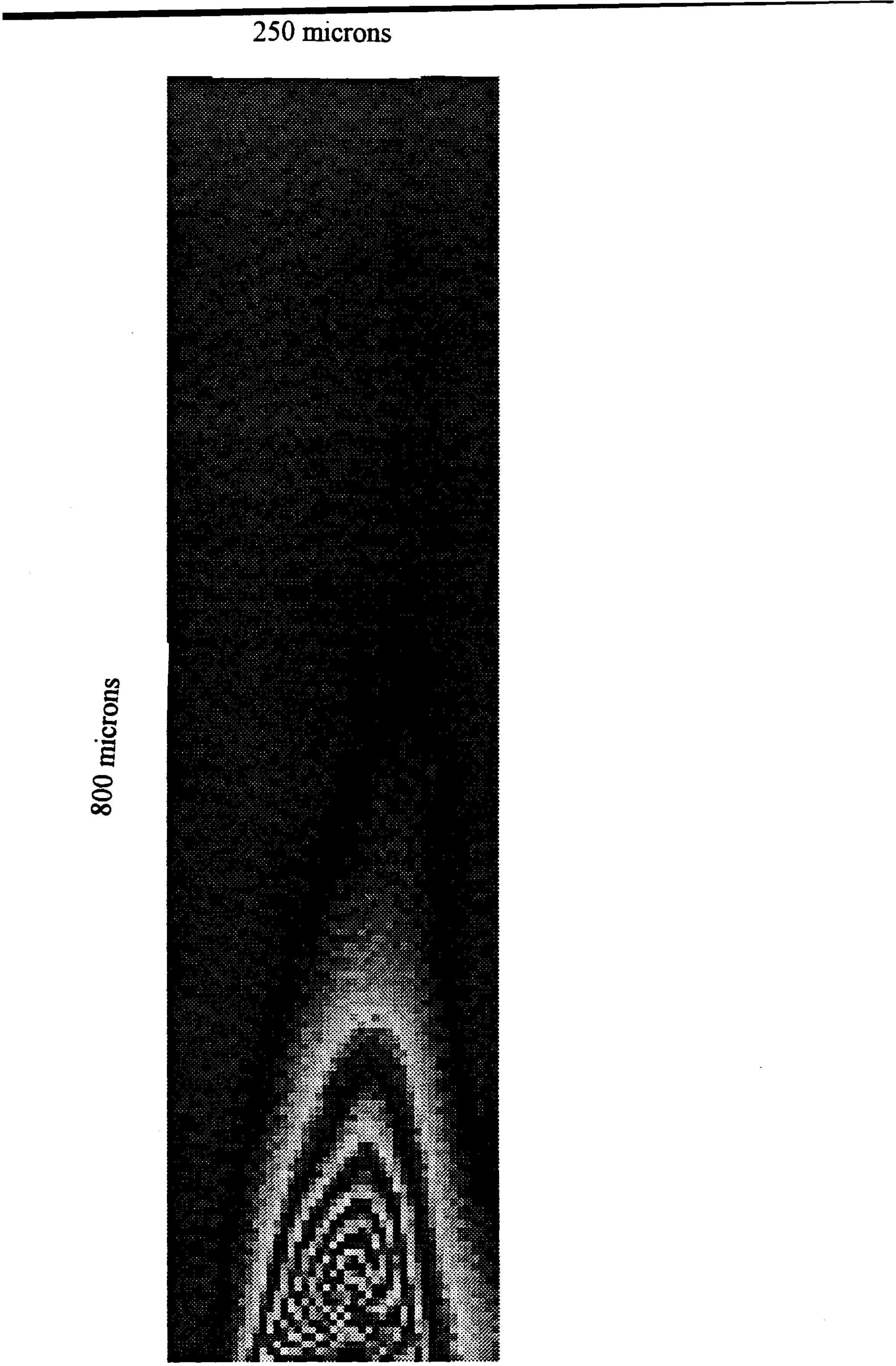


Figure 6.

Low detail scan of undercut ceramic. By using a wide probe beam, the small cracks are blurred out, revealing the major structure beneath.

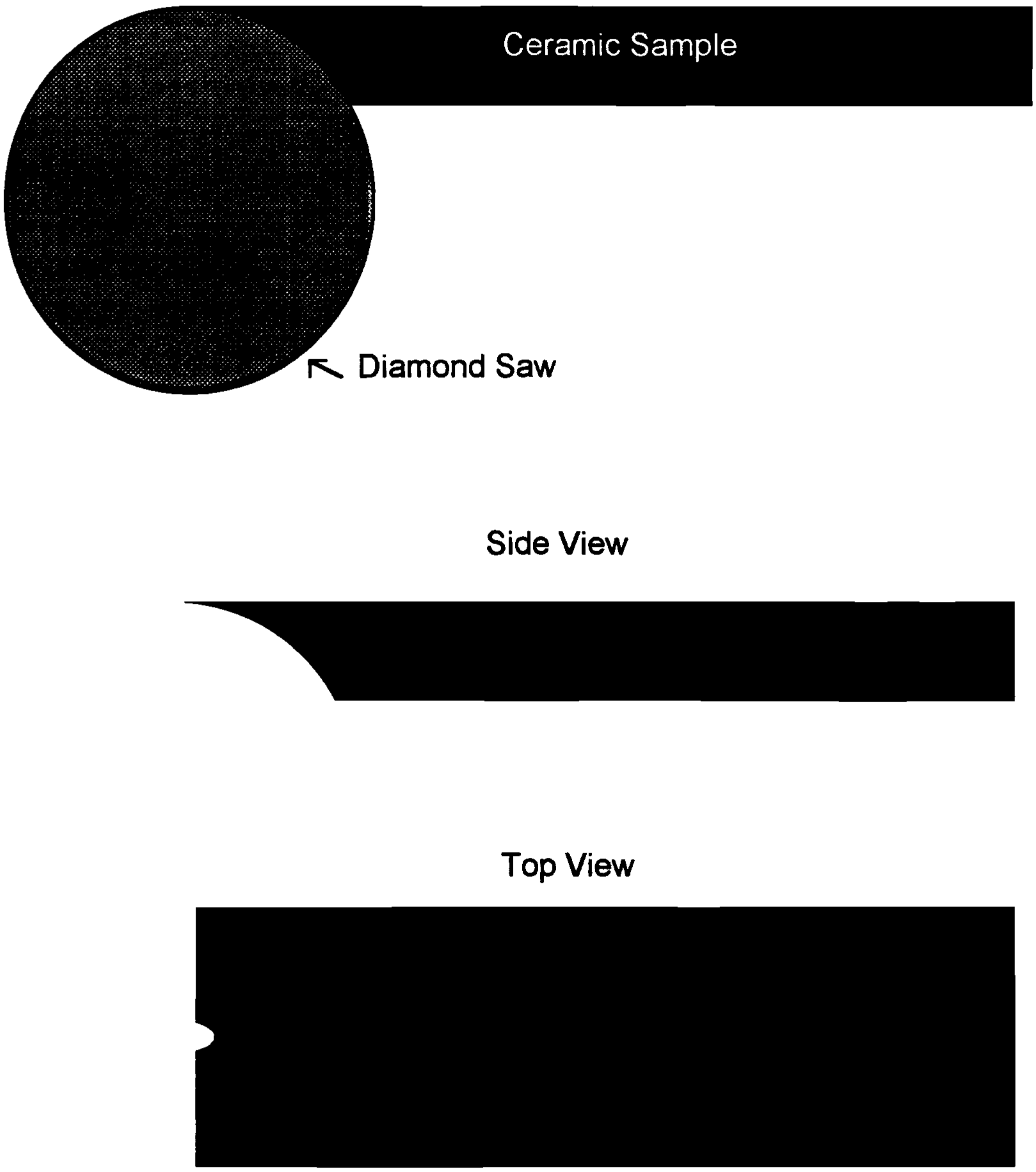


Figure 7.

Orientation of undercut ceramic sample

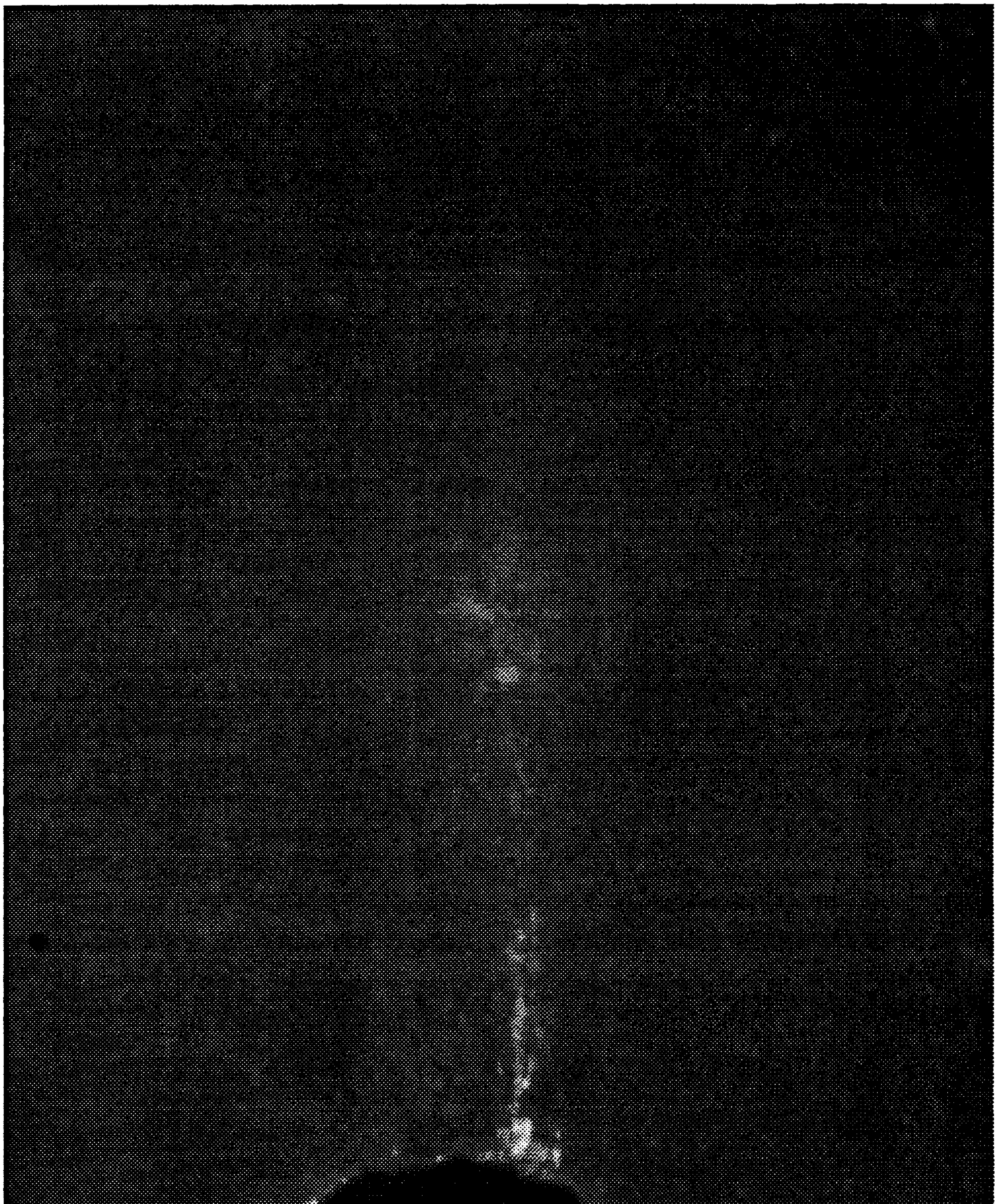


Figure 8

Optical image of the surface of the undercut ceramic. Base image is at 500 times magnification.

ANNEALED AND UNANNEALED CERAMICS

AlliedSignal is a manufacturer of high strength ceramics for specialized applications. As such, they were happy to donate samples for inspection. Figures 9 and 10 are scans of two of these samples of ceramics. Figure 9 is the scan of a medium strength ceramic, while figure 10 is a scan of the same type of ceramic that has been annealed. The annealed sample shows a smooth surface that has gradual changes over its surface. The non-annealed sample shows several outstanding spikes. As these scans were of a small area, larger scans were made (figures 11 and 12). Like the earlier scan (figure 9), figure 11 shows a calm area with several spikes. Figure 12 has the same overall smoothness as its counterpart (figure 10). It does show some random spikes, but these spikes are not as drastic as those that were found on the non-annealed sample. This is consistent with what one would expect from the annealing process: the material is stress relieved and becomes more homogeneous. One can also not that the overall signal strength was higher on the annealed samples, this is consistent with a larger grain size of the base ceramic.

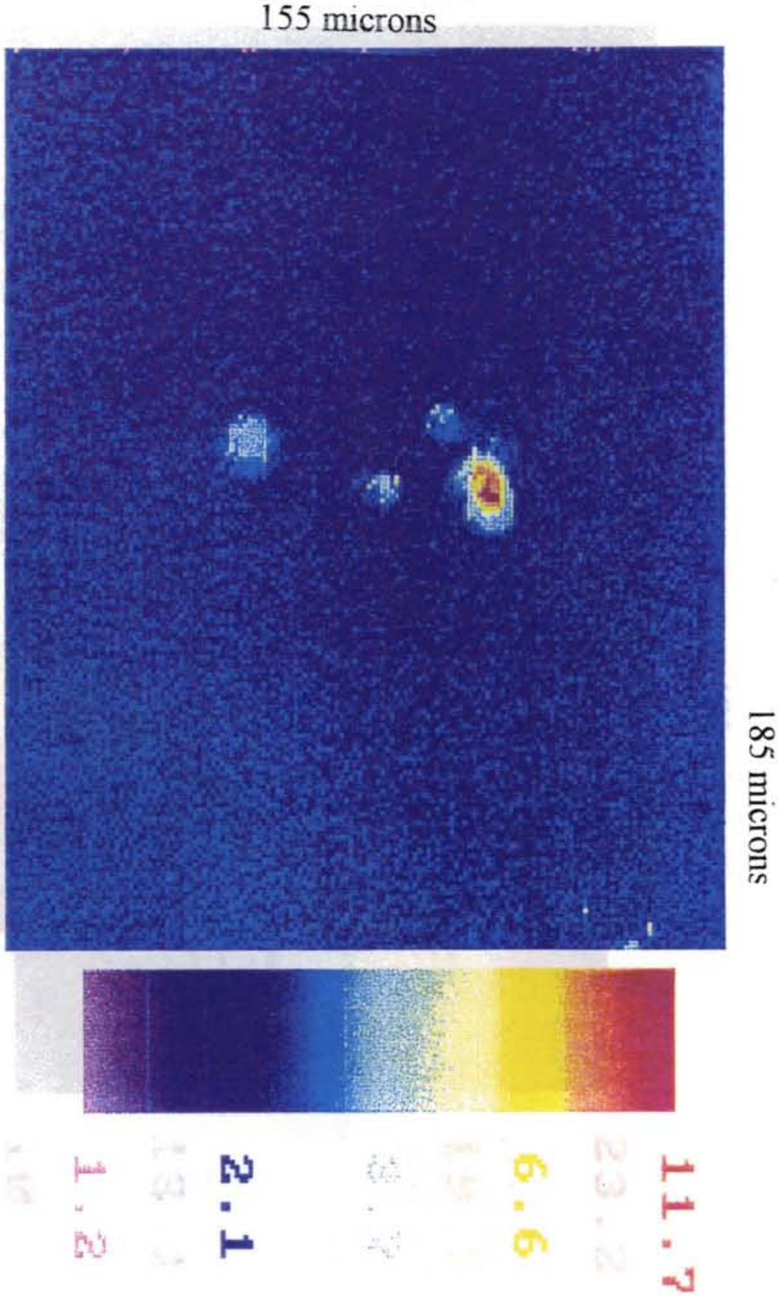


Figure 9.

Non annealed ceramic sample. The peaks are an order of magnitude higher than the floor signal.

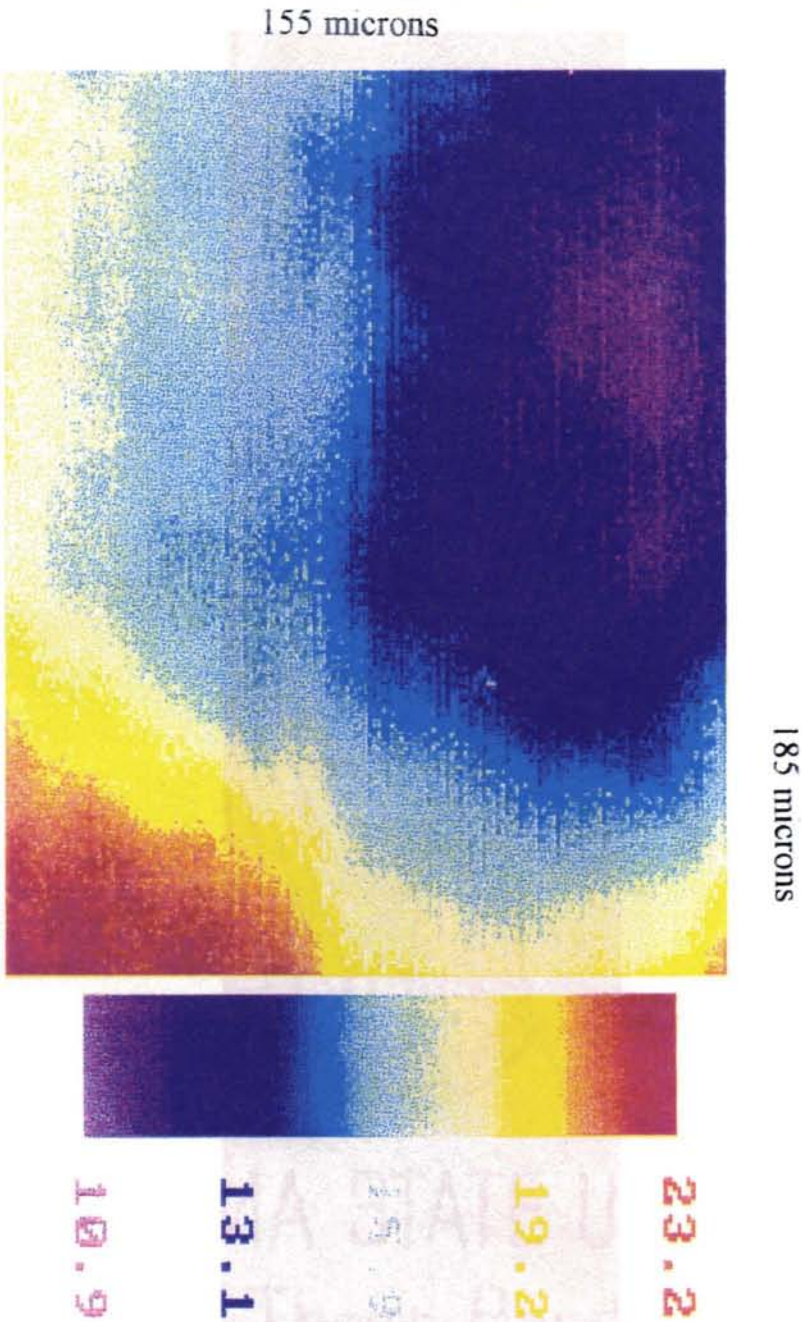


Figure 10.

Annealed ceramic sample. Smooth and consistent response; no major spikes.

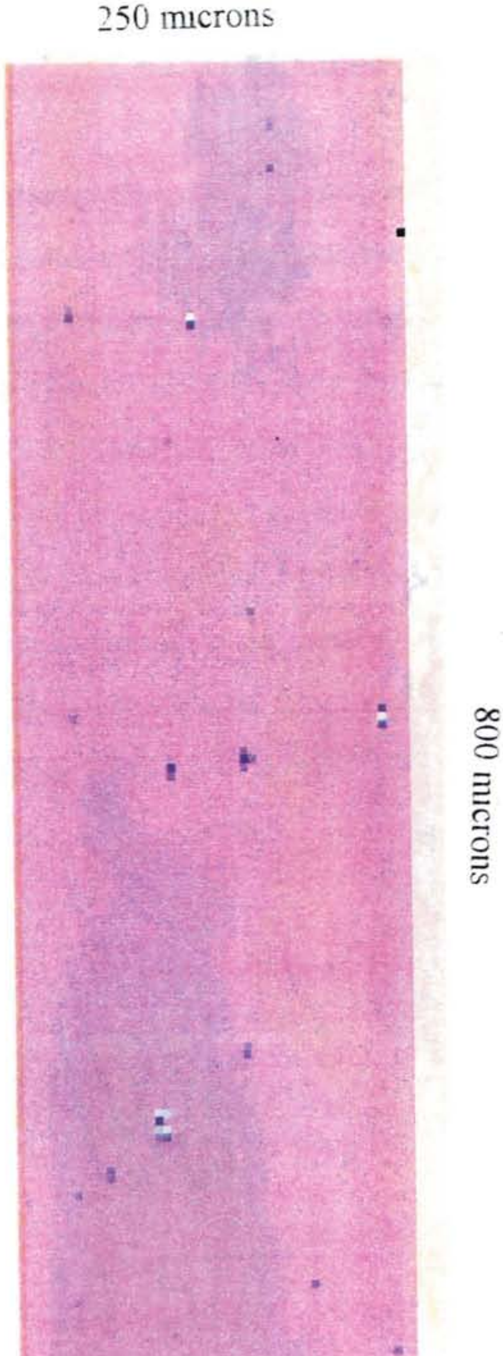


Figure 11

Wide area scan of non annealed ceramic sample. Many random spikes of strong power.

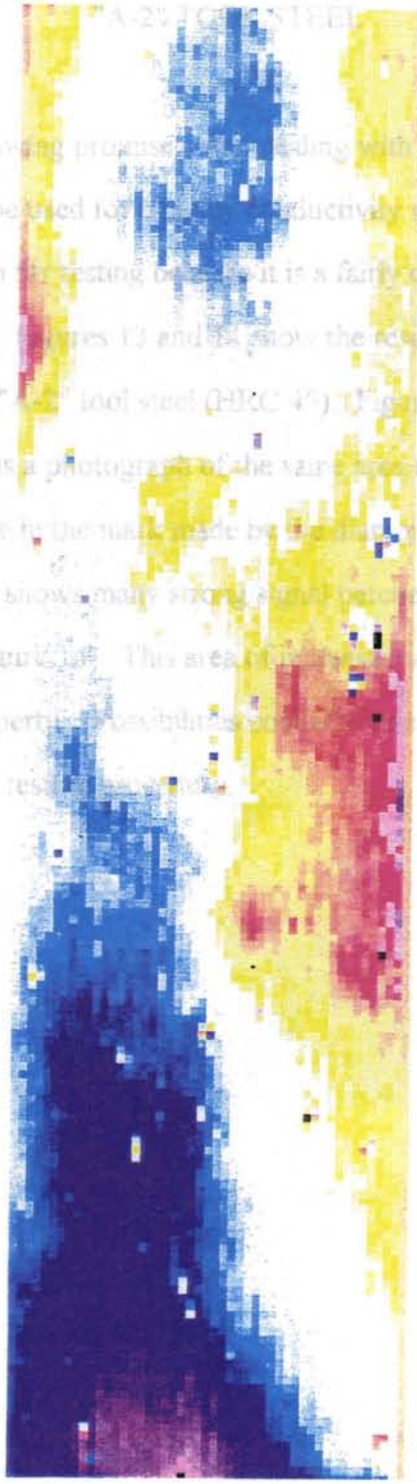


Figure 12

Wide area scan of annealed sample. Spikes that are present are of the same order of magnitude as the floor.

"A-2" TOOL STEEL

The system was showing promise when dealing with ceramics. It was interesting to see if such a system could be used for a highly conductivity material, for example: metal. "A-2" tool steel was chosen for testing because it is a fairly common tool steel that it is frequently used in industry. Figures 13 and 14 show the results of taking a Rockwell "C" hardness test on a piece of "A-2" tool steel (HRC 45). Figure 13 is a photo-thermal scan of the area while figure 14 is a photograph of the same area made with an optical microscope. The large circle in the mark made by the diamond point of the hardness testing machine. Figure 13 shows many strong signal patches just outside the indention zone that don't appear in figure 14. This area of increased variation correspond to radical changes in the material property . Possibilities could include cracks or areas that have been work-hardened by the testing procedure.

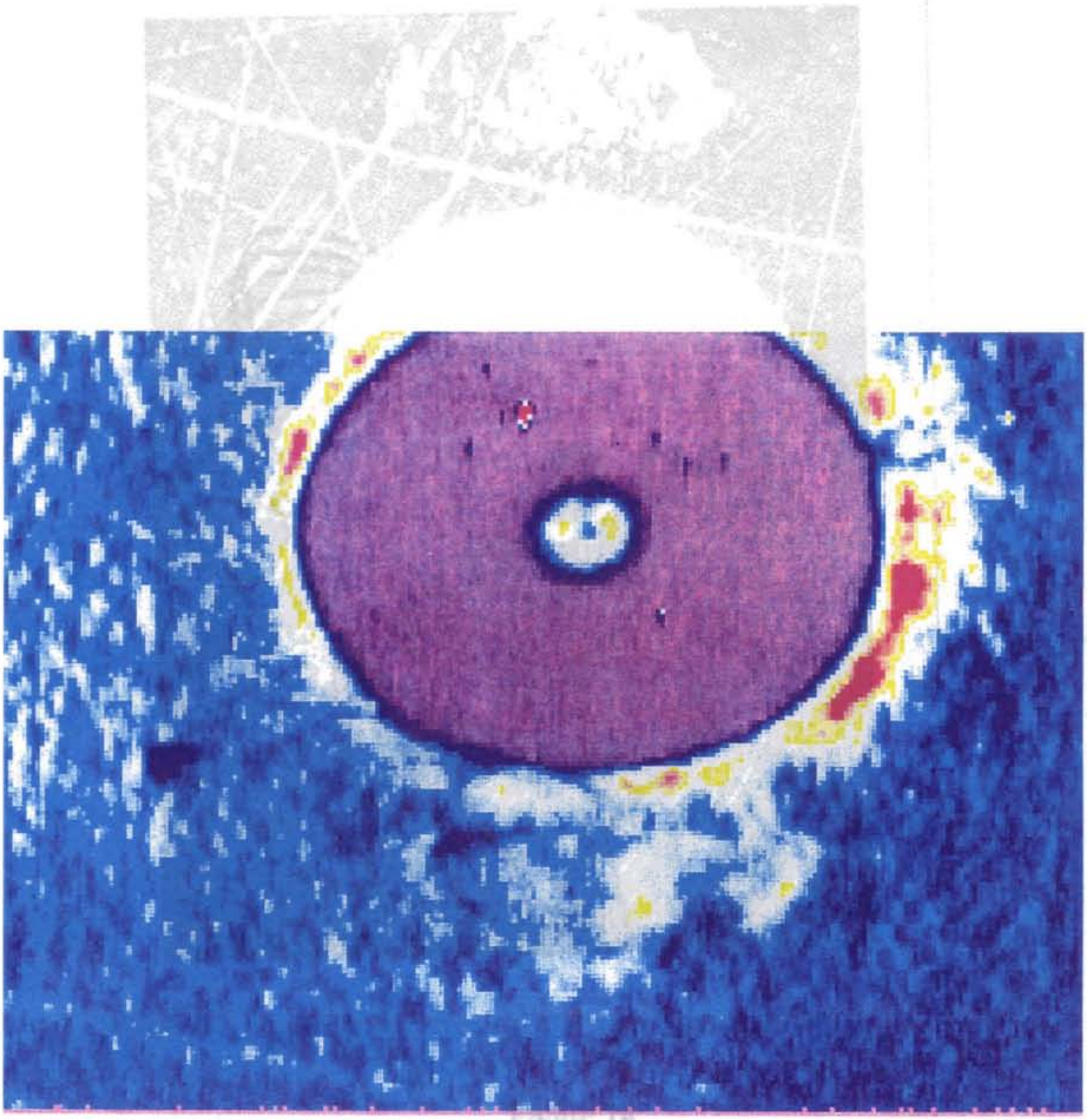


Figure 13

1mm by 1mm area of A-2 tool steel. The circle is the "foot print" of the diamond point used for the Rockwell hardness measurement. Various cracks can be seen along the edge of the imprint zone.

Figure 14

Optical photograph of A-2 tool steel impact zone. Photograph was made at 500 times magnification.

CONTROLLED LOW TEMPERATURE DEPTH TEST

For medical applications, there is interest in profiling materials (skin and organs) in an effort to detect various defects (scar tissue, calcifications, or tumors). In order to gain respectable signal levels at near body temperatures, it was necessary to use a wider spectrum, deeper IR detector. An InSb detector, which is optimum at 3-5 microns, was selected. While this choice in window radically decreased the non linearity of the system, mathematical models showed it should still have a sensitivity on the order of the seventh power. Figures 15, 16, and 17 show the results of a control study. A piece of aluminum was coated with several layers of paint in such a fashion as to create a wedge of paint with a controlled thickness. The figures correspond to a decrease in modulation frequency from 250 Hz to 50 Hz down to 10 Hz. Figure 18 shows the plot of signal versus thickness for these images. As the frequency drops more and more detail can be observed in the system.

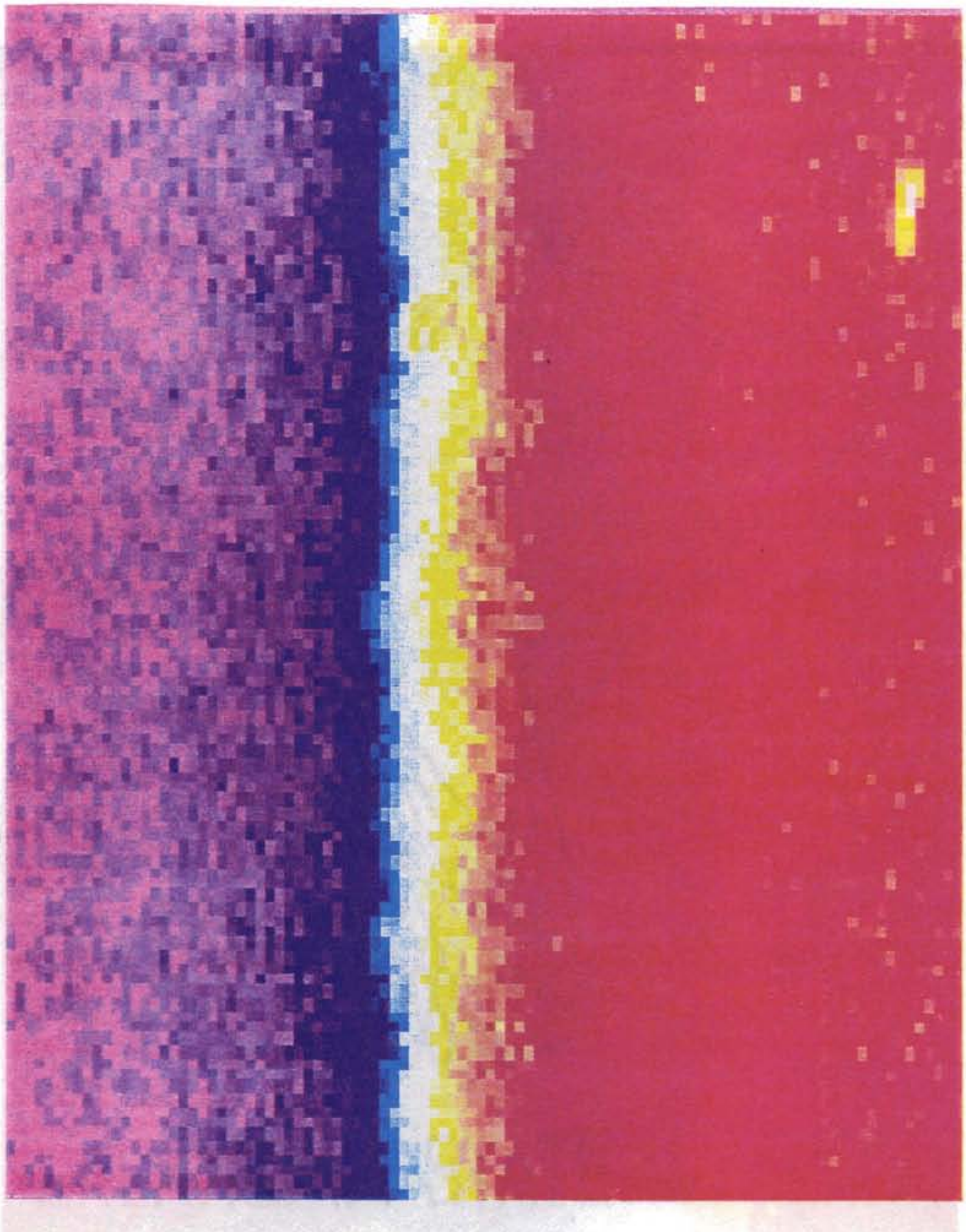


Figure 15

1 cm by 1cm scan of the control paint wedge at 250 Hz. It can be noted that the signal stalls half-way down the image.

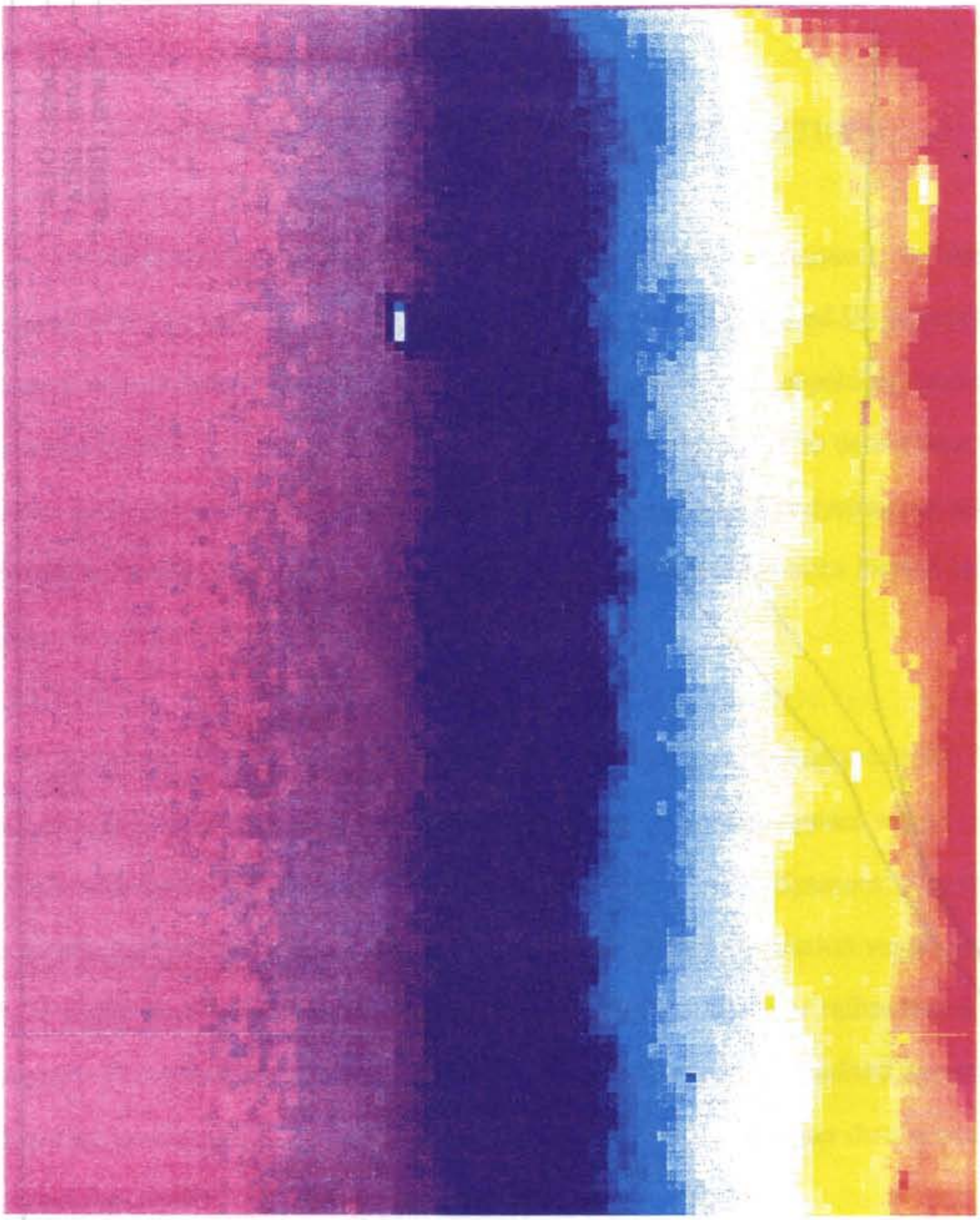


Figure 17

1 cm by 1cm scan of the control paint wedge at 10 Hz. The image shows a smooth gradation over the thickness.

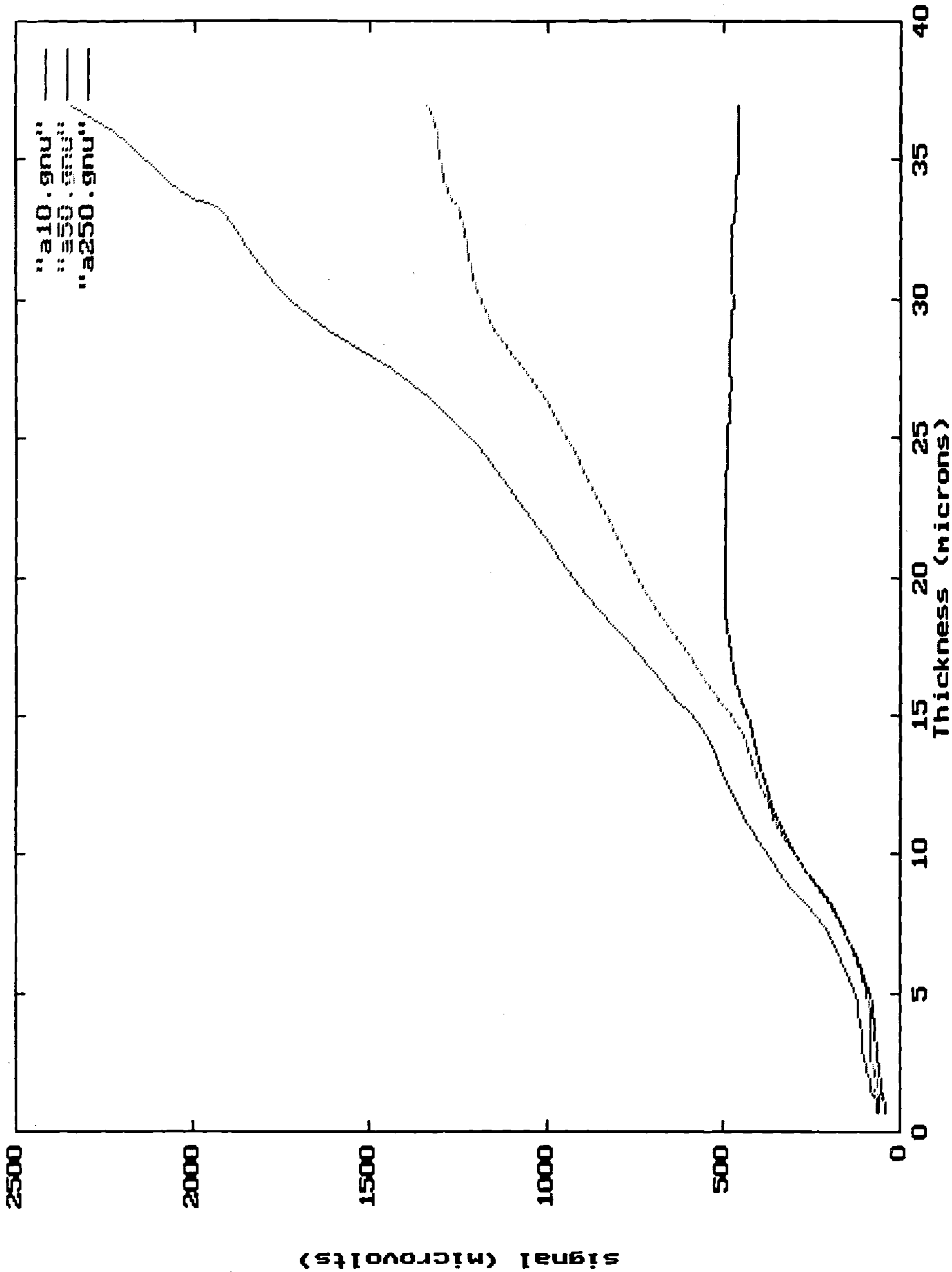


Figure 18

Thickness versus signal comparison for 250, 50 and 10 Hz. The 250 Hz scan flatlines at around 20 microns of depth. The 50 Hz begins to level off at 35 microns while the 10 Hz signal is still going strong.

CHAPTER V.

SUMMARY, CONCLUSION, RECOMMENDATIONS

A photo-thermal nondestructive evaluation system was discussed. By using a chopped laser probe beam and by carefully observation of the emitted IR radiation, the system is able to detect variations in a sample's thermal transport capability, hence-defects. The experimental system as described has demonstrated the ability to detect flaws in various materials, in both the surface and sub-surface regions. The system has also demonstrated high levels of accuracy and repeatability with resolutions limited only by the choice of wavelength for the probe beam.

However, the system as discussed is too slow for industrial applications. Should it be used in the evaluation of ceramic ball bearings, the translation system would not be able to cope with the curve of the ball bearing for more than a few millimeters. The speed of the system can be resolved by using faster processors, faster translation stages, or by using a wide probe source and either a linear or two-dimensional detector rather than a point detector. The curvature problem associated with a sphere can be resolved by using a translation system that can rotate the sample about its own axis rather than around an external axis point.

The system has been shown to work with ceramics and metals. The system has the possibility to work with low temperature items with similar degrees of resolution. As such, it would be worth exploring the use of such a system in the medical field, where this high degree of detection could permit medical personnel to be able to non-intrusively inspect the skin of cancer and burn patients.

REFERENCES

1. Musikant, Solomon. What Every Engineer Should Know About Ceramics, Marcel Dekker, NY 1991.

2. Miller, Gordon H. Photoacoustic Method for Nondestructive Testing of Ceramic Ball Bearings, Masters Thesis, Oklahoma State University, 1994.

3. Thomas, R.L. and Favro, L.D. and Kuo, P.K. Thermal-wave imaging for non-destructive evaluation. Canadian Journal of Physics. no. 64, 1986.

4. Thomas, Favro, Kuo, Ahmed, and Jin. Synchronous Thermal Wave IR Imaging for Nondesctructive Evaluation. Journal of Nondestructive Evaluation, Vol. 8, No. 2, 1989.

5. Aamodt, L.C. and Murphy, J.C. Effect of 3-D heat flow near edges in photothermal measurements. Applied Optics, Vol. 21, No. 1, Jan. 1982

6. Pouch, Thomas, and Wong, Journal of the Optical Society of America, 70, 1980.

7. Tipler, Paul A. Foundations of Modern Physics. Worth Publishers, 1976.

APPENDIX A.

This software was written in QBasic and compiled. Its controls the X-Y translation stage while monitoring the response from the SR-510 via the A to D card. This version is designed to work with two lock-in amplifiers set at 90 degrees of phase to each other. This permits the software to monitor the phase information as well as frees the system from signal fall-off due to phase drift.

Sample numbers from 0-99 are valid.

<esc> sends the translation stage to the "home" position

<g> sends the translation stage to the target position

<q> quits to DOS

<?> starts a scan. Once running <esc> aborts the scan

```
DECLARE SUB LasPow ()

CLS : INPUT "Sample Number:"; nm
addr = &H260
OUT addr + 0, 0
OUT addr + 1, 0
OUT addr + 2, 0
OUT addr + &H17, &H34
OUT addr + &H14, &HA
OUT addr + &H14, 0
OUT addr + &HA, 0
OUT addr + &HC, 0
OUT addr + &H8, 0
a = INP(addr + &HA)
a = INP(addr + &HA)

DEFDBL X-Y
tft = 200
DIM gain(24): FOR a = 1 TO 24: READ gain(a): NEXT
DATA 10,20,50,100,200,500,1,2,5,10,20,50,100,200,500,1,2,5,10,20,50,100,200,500

DEF fncv (r, g, b) = INT(r) + 256 * INT(g) + 65536 * INT(b)
```

```
DIM pal&(255)
DIM info(100), phase(100)

SCREEN 13

FOR x = 0 TO 255
 LINE (x, 0)-(x, 100), x
NEXT x

p = 1
r = 63
FOR x = 0 TO 63
 g = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 1 TO 63
 r = 63 - x
 b = x / 2
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 b = 32 + x / 2
 g = 63 - x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 r = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x




DIM noise(30), raw(30)
DIM sortit(63, 2)
nm$ = STR$(nm)
OPEN "area" + MID$(nm$, 2, LEN(nm$) - 1) + ".iii" FOR OUTPUT AS #2

IBMBASE = &HD800
IBMBASE$ = "D800"
CONST data.out = &H3FC
CONST data.in = &H3FE
CONST data.avail = &H800
```

```
DEF SEG = IBMBASE     'SET BASE ADDRESS TO MATCH BOARD ADDRESS
GOSUB dumb

OPEN "com2:19200,n,8,2,cs,ds,cd" FOR RANDOM AS #1
PRINT #1, "   "
PRINT #1, "G"
INPUT #1, g: sr510gain = gain(g): PRINT #2, g
PRINT #1, "P": INPUT #1, phase: PRINT #2, phase
PRINT #1, "F": INPUT #1, freq: PRINT #2, freq
forward = .25
backward = -.2

xtarget = 0
ytarget = 0
xlocation = 0
ylocation = 0
walk = 0
speed = 0


CLS


GOSUB GoOn

xtarget = 0: ytarget = 0
'GOSUB GoAbsolute
 GOSUB dumb
  stuff$ = "SG50": GOSUB SendIt: GOSUB dumb: stuff$ = "SE255": GOSUB SendIt
 GOSUB dumb
 stuff$ = "SA8000000": GOSUB SendIt: GOSUB dumb

 stuff$ = "SD8000000": GOSUB SendIt: GOSUB dumb
 stuff$ = "SP1": GOSUB SendIt: GOSUB dumb

RMSvalue = 0: rms = 0
speed = 500: GOSUB GoSpeed
GOSUB HowFast
tm = TIMER + 1
walk = 0
peat:

 a$ = INKEY$: IF a$ = " " THEN walk = 0
 IF a$ = CHR$(27) THEN GOSUB GoHome
 IF a$ = "g" THEN GOSUB GoAbsolute

 IF a$ = "6" THEN xtarget = xtarget + 8.3333
 IF a$ = "4" THEN xtarget = xtarget - 8.3333
 IF a$ = "8" THEN ytarget = ytarget + 8.3333
 IF a$ = "2" THEN ytarget = ytarget - 8.3333

 IF a$ = "f" THEN speed = 2000: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "s" THEN speed = 500: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "F" THEN speed = 5000: GOSUB GoSpeed: GOSUB HowFast
```

```
 IF a$ = "S" THEN speed = 100: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "q" THEN END
 IF a$ = "?" THEN GOTO SnarfData
 IF TIMER > tm THEN tm = TIMER + 1: GOSUB Stats

GOTO peat

SR510v:
  PRINT #1, "Q"
  INPUT #1, volt
  v = volt / (1E-09)
  ap$ = "n": IF v > 1000 THEN v = v / 1000: ap$ = "æ"
  IF v > 1000 THEN v = v / 1000: ap$ = "m"
  v = INT(v * 1000 + .5) / 1000
  volt$ = STR$(v) + " " + ap$ + "V"
  RETURN



GETd812:
stuff = 0
dead1:
  IF (PEEK(data.avail) AND 2) = 0 THEN GOTO dead1
  stuff = PEEK(data.in)
  'PRINT stuff, CHR$(stuff)
RETURN

GoHome:
  stuff$ = "GH": GOSUB SendIt
RETURN

GoOn:
  stuff$ = "MN": GOSUB SendIt
RETURN

GoAbsolute:
  xtarget = CLNG(xtarget + .5)
  ytarget = CLNG(ytarget + .5)

  stuff$ = "3MA" + STR$(xtarget)
  GOSUB SendIt
  stuff$ = "4MA" + STR$(ytarget)
  GOSUB SendIt
RETURN

GoSpeed:
  GOSUB dumb
  stuff$ = "SV" + STR$(speed)
  GOSUB SendIt
RETURN

GoSpeedX:
  GOSUB dumb
  stuff$ = "3SV" + STR$(speed)
  GOSUB SendIt
```

```
RETURN

GoSpeedY:
 GOSUB dumb
 stuff$ = "4SV" + STR$(speed)
 GOSUB SendIt
RETURN


SendIt:
'PRINT "sendit:"; stuff$
 GOSUB dumb
 stuff$ = stuff$ + CHR$(13)
 count = LEN(stuff$)
 FOR s = 1 TO count
  a$ = MID$(stuff$, s, 1)
  stuff = ASC(a$)
retry:
  IF (PEEK(data.avail) AND 1) = 0 THEN
   POKE data.out, stuff
   POKE data.out + 3, stuff
   'PRINT "send "; CHR$(stuff), stuff
  ELSE
   GOSUB dumb
   GOTO retry
  END IF

 NEXT
RETURN

HowFast:
 GOSUB dumb
 stuff$ = "3TY": GOSUB SendIt
 a$ = ""
dead9:
  GOSUB GETd812
  IF stuff <> ASC("Y") THEN GOTO dead9
dead10:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead10
  speed = VAL(a$)
  GOSUB dumb
RETURN

WhereAmI:
  stuff$ = "3TP": GOSUB SendIt
  a$ = ""
dead3:
  GOSUB GETd812
  IF stuff <> ASC("P") THEN GOTO dead3
dead4:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead4
```

```
  xlocation = VAL(a$)
  GOSUB dumb
  stuff$ = "4TP": GOSUB SendIt
 a$ = ""
dead5:
  GOSUB GETd812
  IF stuff <> ASC("P") THEN GOTO dead5
dead6:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead6
  GOSUB dumb
  ylocation = VAL(a$)
RETURN

dumb:
dead0:
 IF (PEEK(data.avail) AND 2) <> 0 THEN a = PEEK(data.in): GOTO dead0
 ' PRINT a
RETURN

AreWeDone:
  stuff$ = "3TS": GOSUB SendIt
  a$ = ""
dead11:
  GOSUB GETd812
  IF stuff <> ASC("S") THEN GOTO dead11
dead12:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead12
  st1 = VAL(a$)
  GOSUB dumb
  stuff$ = "4TS": GOSUB SendIt
 a$ = ""
dead13:
  GOSUB GETd812
  IF stuff <> ASC("S") THEN GOTO dead13
dead14:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead14
  GOSUB dumb
  st2 = VAL(a$)
  st3 = 0
  IF (st1 AND 1) AND (st2 AND 1) THEN st3 = 1
RETURN

Stats:
 GOSUB WhereAmI
 LOCATE 1, 1: PRINT USING "X-Loc: ####.###    Y-Loc: ####.###"; xlocation * .06; ylocation * .06
 PRINT USING "X-Tar: ####.###    Y-Tar: ####.###"; xtarget * .06; ytarget * .06
 PRINT USING "Speed: #####"; speed
 GOSUB SR510v
 PRINT "Voltage: " + volt$ + "        "
```

```
'GOSUB AreWeDone: PRINT USING "ST1 ###   ST2 ###  ST3 #"; st1; st2; st3


rms = rms + 1
raw(rms) = volt
IF rms < 30 THEN GOTO gondi

q = 0: FOR a = 1 TO 30: q = q + raw(a): NEXT a: q = q / 30
FOR a = 1 TO 30: noise(a) = raw(a) - q: NEXT a: ref = q
q = 0: FOR a = 1 TO 30: q = q + noise(a) * noise(a): NEXT

q = SQR(q / 30)

q = (q / ref) * 100
LOCATE 11, 10: PRINT USING "RMS Noise:###.# %"; q
rms = 29
FOR a = 1 TO 29: raw(a) = raw(a + 1): NEXT a
gondi:
RETURN

SnarfData:
 CLS
 sttime = TIMER
 counter = 0
 PRINT "1"

  xtarget = -50: ytarget = 0
 GOSUB GoAbsolute
here:
 GOSUB WhereAmI
 IF xlocation < (-20) THEN GOTO here
   speed = 5000: GOSUB GoSpeedY
  speed = 5000: GOSUB GoSpeedX



 odd = 0

 vmax = 0: vmin = 4096
  CLS
 sttime = TIMER
 alt = 0

  speed = 15000: GOSUB GoSpeedX
  xtarget = -50: GOSUB GoAbsolute
check3:
  GOSUB WhereAmI
  IF ABS(xlocation - xtarget) > 30 THEN GOTO check3



  FOR ytag = -5000 TO 5000 STEP 100
  ytarget = CLNG(ytag / .06)
  GOSUB GoAbsolute
check2:
```

```
   GOSUB WhereAmI
   IF ABS(ylocation - ytarget) > 30 THEN GOTO check2



   speed = 45000: GOSUB GoSpeedX
   xtarget = 170000: tag = 1
   IF odd = 1 THEN xtarget = -500
   GOSUB GoAbsolute
check:

GOSUB WhereAmI

   volt = 0: phs = 0
   FOR vcount = 1 TO tft: CALL LasPow: volt = volt + 1024 * tlaspow
   GOSUB atan2
   phs = phs + angle
   NEXT: volt = CLNG(volt / tft): phs = CLNG(phs / tft * 10)

   IF volt > vmax THEN vmax = volt
   IF volt < vmin THEN vmin = volt
    xx = CLNG(((xlocation * .06)) / 100)
    yy = CLNG(((ylocation * .06) + 5000) / 100)
    xxx = xx * 2
    yyy = yy * 2
    quit$ = INKEY$: IF quit$ = CHR$(27) THEN GOTO OldMan
     IF xx > 0 AND xx < 100 THEN
       info(xx) = volt
       phase(xx) = phs
       LINE (xxx + 1, yyy + 1)-(xxx, yyy), 256 - volt, B
       IF volt > vmax THEN vmax = volt
       IF volt < vmin THEN vmin = volt
     END IF
    IF (odd = 0) AND (xlocation < 167000) THEN GOTO check
    IF (odd = 1) AND (xlocation > 0) THEN GOTO check

   speed = 150000: GOSUB GoSpeed
   odd = odd + 1: IF odd > 1 THEN odd = 0
   FOR we = 1 TO 100
    PRINT #2, info(we): info(we) = 0
    PRINT #2, phase(we): phase(we) = 0
    NEXT

   ytarget = ytarget - 150
   GOSUB GoAbsolute

check6:
   GOSUB WhereAmI
   IF SQR((ylocation - ytarget) ^ 2) > 20 THEN GOTO check6



    NEXT ytag
```

```
OldMan:
   PRINT #2, 0, 0, -999
   PRINT #2, vmax, vmiv
CLOSE 2
CLOSE 1
 PRINT TIMER - sttime
 PRINT vmax, vmin
GOSUB GoHome:
a$ = INPUT$(2)
END

atan2:
 ap = ABS(laspow1)
 bp = ABS(laspow2)
 a = laspow1
 b = laspow2
 IF (a = 0) AND (b = 0) THEN RETURN
 power = SQR(a ^ 2 + b ^ 2)

 IF a > 0 AND b > 0 THEN q = 0
 IF a < 0 AND b > 0 THEN q = 1
 IF a < 0 AND b < 0 THEN q = 2
 IF a > 0 AND b < 0 THEN q = 3

 IF bp >= ap THEN
    eep = 0
    angle = 90 - rtd * ATN(a / b)
     IF q > 1 THEN angle = angle + 180

   END IF
 IF ap > bp THEN
     eep = 1
     angle = rtd * ATN(b / a)
     IF q > 0 THEN angle = angle + 180
     IF q = 3 THEN angle = angle + 180
   END IF

RETURN

SUB LasPow
SHARED addr, laspow1, laspow2, tlaspow
 s = INP(addr)
 OUT addr, 0
 OUT addr + 3, 0
piddy:
 s = INP(addr)
 IF (s AND 1) = 0 THEN GOTO piddy
 lpow = INP(addr + &HA)
 hpow = INP(addr + &HA)
 laspow1 = (hpow * 256 + lpow - 2048)

 s = INP(addr)
 OUT addr, 1
 OUT addr + 3, 0
piddy2:
```

```
  s = INP(addr)
  IF (s AND 1) = 0 THEN GOTO piddy2
  lpow = INP(addr + &HA)
  hpow = INP(addr + &HA)
  laspow2 = (hpow * 256 + lpow - 2048)

tlaspow = ((laspow1 ^ 2 + laspow2 ^ 2) ^ .5) / 4096
END SUB
```

Appendix B.

This software was written in QBasic and compiled. It takes the ".iii" files generated from the scaning program and compresses them, and stamps them with a version number, time/date, gain, phase and frequency information (pertaining to the settings on the SR-510 lock-in amplifier).

```
CLS
INPUT "Sample:"; nm$
DIM q(100, 100) AS INTEGER
DIM w(100, 100) AS INTEGER
vmx = -10000: vmn = 10000


OPEN "area" + MID$(nm$, 1, LEN(nm$)) + ".iii" FOR INPUT AS #1
OPEN "area" + MID$(nm$, 1, LEN(nm$)) + ".wlf" FOR OUTPUT AS #2

 INPUT #1, gain
 INPUT #1, phase
 INPUT #1, freq

FOR y = 1 TO 100
 FOR x = 1 TO 100
 INPUT #1, q(x, y)
 INPUT #1, w(x, y)
 IF q(x, y) > vmx THEN vmx = q(x, y)
 IF q(x, y) < vmn THEN vmn = q(x, y)
 NEXT
NEXT



 PRINT #2, "WOLF"
 PRINT #2, 3.1
 PRINT #2, DATE$
 PRINT #2, gain
 PRINT #2, phase
 PRINT #2, freq
 PRINT #2, vmx
 PRINT #2, vmn

FOR y = 1 TO 100
FOR x = 1 TO 100
    v = q(x, y) + 2048
    hs = INT(v / 256)
    ls = v MOD 256
```

```
      PRINT #2, CHR$(hs);
      PRINT #2, CHR$(ls);

    v = w(x, y)
    hs = INT(v / 256)
    ls = v MOD 256

      PRINT #2, CHR$(hs);
      PRINT #2, CHR$(ls);

NEXT: NEXT
```

Apendix C.

This software was written in QBasic. It was not compiled due to the fact that the supplied compiler could not deal with large memory models. This program will load up to three different images. It can display images side by side, or in a diferential form. Indvidual images may be displayed full screen with gradiated index on the side. This program expects to load ".wlf" files (Appendix B covers the ".iii" to ".wlf" software, Apendix D. discusses how these files are encoded)

```
DEF fncv (r, g, b) = INT(r) + 256 * INT(g) + 65536 * INT(b)
DIM pal&(255)
DIM a(100, 100, 2) AS INTEGER
DIM files$(2)
SCREEN 13
CLS
FOR x = 0 TO 255
 LINE (x, 0)-(x, 10), x
NEXT x

p = 1
r = 63
FOR x = 0 TO 63
 g = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 1 TO 63
 r = 63 - x
 b = x / 2
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 b = 32 + x / 2
 g = 63 - x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 r = x
 pal&(p) = fncv(r, g, b)
```

```
 PALETTE p, pal&(p)
 p = p + 1
NEXT x
p = 0: r = 63: g = 63: b = 63: pal&(p) = fncv(r, g, b): PALETTE p, pal&(p)


menuarea:
 rover = 1
 nf = 0
 dif = 0
peatmenu:
CLS
COLOR 200
 PRINT "1) "; : IF nf = 0 THEN COLOR 20: PRINT "<none>" ELSE PRINT file$(0)
COLOR 200
 PRINT "2) "; : IF nf < 2 THEN COLOR 20: PRINT "<none>" ELSE PRINT file$(1)
 PRINT : PRINT
COLOR 200
 PRINT "3) ";
 IF nf <> 2 THEN dif = 0
 IF (dif = 0) AND (nf < 3) THEN COLOR 20: PRINT "<none>"
 IF dif = 0 AND nf = 3 THEN PRINT file$(2)
 IF (dif = 1) AND nf = 2 THEN COLOR 120: PRINT " #1 vs #2 "
COLOR 200


 PRINT : PRINT :
 PRINT "A)dd File"
 PRINT "R)emove"
 PRINT "D)ifference"
 PRINT "P)lot All"
 PRINT "E)xpanded: "; : IF rover = 2 THEN PRINT "ON" ELSE PRINT "OFF"
 PRINT "<ESC> Reset"
 PRINT "Q)uit"


 i$ = INPUT$(1)
 IF (i$ = "d" OR i$ = "D") AND (nf = 2) THEN
        dif = dif + 1
        IF dif = 1 THEN GOSUB MakeDif
      END IF
 IF dif > 1 THEN dif = 0
 IF (i$ = "a" OR i$ = "A") AND (nf < 3) THEN
           CLS
           INPUT "Name:"; file$(nf)
           n = nf: nm$ = file$(n)
           GOSUB GetIt
           nf = nf + 1
         END IF
 IF i$ = "r" OR i$ = "R" THEN nf = nf - 1
 IF (i$ = "p" OR i$ = "P") AND (nf > 0) THEN
          CLS
          FOR q = 1 TO nf
           n = q - 1
           GOSUB PlotIt
```

```
            NEXT
            IF dif = 1 THEN n = 2: GOSUB PlotIt
            i$ = INPUT$(1)
          END IF
  IF i$ = "q" OR i$ = "Q" THEN END
  IF i$ = CHR$(27) THEN GOTO menuarea
  IF i$ = "e" OR i$ = "E" THEN rover = rover + 1: IF rover > 2 THEN rover = 1
  i = VAL(i$)
  IF i > 0 AND (nf + dif) >= i THEN
    CLS
    n = i - 1
    GOSUB PlotItBig:
    i$ = INPUT$(1)
    IF i$ = CHR$(13) THEN GOSUB Off2Disk
    END IF




  GOTO peatmenu

'nm$ = "area95": n = 0: GOSUB GetIt: GOSUB PlotIt
'nm$ = "area97": n = 1: GOSUB GetIt: GOSUB PlotIt
'GOSUB PlotDif


END
MakeDif:
vmn(2) = 10000: vmx(2) = -10000
rng0 = vmx(0) - vmn(0): rng1 = vmx(1) - vmn(1): scl = rng0 / rng1
  file$(2) = "#1 vs #2"
  vers(2) = 0
  FOR y = 1 TO 100
  FOR x = 1 TO 100
   v1 = a(x, y, 1) * scl: v0 = a(x, y, 0)
   d = v1: IF v0 < v1 THEN d = v0
   IF d = 0 THEN d = 1
   v = (ABS((v1 - v0) / d) * 196) MOD 254
   IF v > vmx(2) THEN vmx(2) = v
   IF v < vmn(2) THEN vmn(2) = v
   a(x, y, 2) = v
  NEXT
  NEXT
RETURN

PlotIt:
COLOR 254
LOCATE 1, 1 + n * 13: PRINT file$(n);

IF vers(n) > 0 THEN
 LOCATE 20, 1 + n * 13: PRINT d$(n);
 IF vers(n) > 1 THEN
    LOCATE 23, 1 + n * 13: PRINT "Ph :"; phase(n);
    LOCATE 24, 1 + n * 13: PRINT "Frq:"; freq(n);
```

```
  END IF
END IF
 stretch = rover * 254 / (vmx(n) - vmn(n))
  FOR y = 1 TO 100
  FOR x = 1 TO 100
   v = a(x, y, n) - vmn(n)
   IF vers(n) < 0 THEN v = v * .25 + 128
   vv = 256 - v * stretch: IF vv > 254 THEN vv = 254
   IF vv < 1 THEN vv = 1
   PSET (x + 100 * n, y + 10), vv
  NEXT
  NEXT
RETURN

PlotItBig:
COLOR 254
 stretch = rover * 254 / (vmx(n) - vmn(n))
  FOR y = 1 TO 100
  FOR x = 1 TO 100
   v = a(x, y, n) - vmn(n)
   IF vers(n) < 0 THEN v = v * .25 + 128
   vv = 256 - v * stretch: IF vv > 254 THEN vv = 254
   IF vv < 1 THEN vv = 1

   yy = y * 2 - 2
   xx = x * 2.5 - 2
   LINE (xx, yy)-(xx + 3, yy + 2), vv, BF
  NEXT
  NEXT
  FOR y = 1 TO 254: LINE (253, y * .78)-(263, y * .78), y: NEXT
  FOR sc = 0 TO 1 STEP .25
   LOCATE 24 * (1 - sc) + 1, 35
   COLOR 255 - 254 * sc
   PRINT sc;
  NEXT
RETURN

GetIt:
OPEN nm$ + ".wlf" FOR BINARY AS 1
T$ = INPUT$(6, 1)
GOSUB whip: vers(n) = z
i$ = INPUT$(1, 1)
d$(n) = INPUT$(11, 1)
GOSUB whip: g = z
IF vers(n) > 1 THEN
    GOSUB whip: phase(n) = z
    GOSUB whip: freq(n) = z
  END IF
GOSUB whip: vmx(n) = z
GOSUB whip: vmn(n) = z
'PRINT vmn, vmx, g, vers, d$: STOP
i$ = INPUT$(1, 1)
FOR y = 1 TO 100
  FOR x = 1 TO 100
  hs = ASC(INPUT$(1, 1))
```

```
  ls = ASC(INPUT$(1, 1))
   v = hs * 256 - 2048 + ls
   a(x, y, n) = v
 IF vers(n) >= 3 THEN
      hs = ASC(INPUT$(1, 1))
      ls = ASC(INPUT$(1, 1))
      v = hs * 256 + ls
      IF v > 180 THEN v = v - 360
      IF vers(n) > 3 THEN v = v / 10
      IF nf < 2 THEN a(x, y, n + 1) = v
    END IF
  NEXT
NEXT
CLOSE 1
v = 0: FOR y = 10 TO 90: v = v + a(y, 1, n): NEXT: vmn(n) = v / 80
v = 0: FOR y = 10 TO 90: v = v + a(y, 99, n): NEXT: vmx(n) = v / 80

IF vers(n) >= 3 AND nf < 2 THEN
     nf = nf + 1
     file$(nf) = file$(n) + " phs"
     vers(nf) = -1
     vmx(nf) = 180
     vmn(nf) = -180
    END IF
RETURN
whip:
 x$ = ""
ick:
  z$ = INPUT$(1, 1): IF z$ = CHR$(13) THEN RETURN
  IF z$ = "-" THEN x$ = "-"
  IF z$ = "." THEN x$ = x$ + "."
  IF (VAL(z$) = 0 AND z$ <> "0") THEN GOTO ick
  x$ = x$ + z$
  z = VAL(x$)
  GOTO ick

Off2Disk:
OPEN file$(n) + ".red" FOR OUTPUT AS #1
OPEN file$(n) + ".grn" FOR OUTPUT AS #2
OPEN file$(n) + ".blu" FOR OUTPUT AS #3
FOR y = 0 TO 199
FOR x = 0 TO 319
 c = POINT(x, y): p = pal&(c)
 b = 4 * INT(p / 65536): p = p MOD 65536
 g = 4 * INT(p / 256): r = 4 * (p MOD 256)
PRINT #1, CHR$(r);
PRINT #2, CHR$(g);
PRINT #3, CHR$(b);
NEXT
NEXT
CLOSE 3, 2, 1
RETURN
```

Appendix D.

".WLF" Encoding Scheme

The filename is expected to be "<NAME>.WLF".

HEADER FORMAT

Field	ID	Format	Size	Versions
header	WOLF<CRLF>	ASCII	6 bytes	1.0+
version	#.#<CRLF>	ASCII	3+ bytes	1.0+
date	<as per DOS>	ASCII	11 bytes	1.0+
gain	#(1-24)<CRLF>	ASCII	3+ bytes	1.0+
phase	#.#<CRLF>	ASCII	3+ bytes	2.0+
frequency	#.#<CRLF>	ASCII	3+ bytes	2.0+
max value	#<CRLF>	ASCII	3+ bytes	1.0+
min value	#<CRLF>	ASCII	3+ bytes	1.0+

DATA FORMAT

versions 1-3.5 are of a 100 by 100 array

(all versions) VALUE<High Byte><Low Byte>

(version 3.0+) PHASE<High Byte><Low Byte>

repeat until end.

for example a version 3.0 data file would be as:

<value1.1><phase1.1><value1.2><phase1.2> ... <value100.100><phase100.100>

VITA

Marcus Jeanfreau la Grone

Candidate for the Degree of

Master of Science

Thesis: A PHOTO-THERMAL NON-DESTRUCTIVE EVALUATION METHOD FOR CERAMICS WITH APPLICATIONS FOR OTHER MATERIALS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born at Grissom Air Force Base, Indiana, on October 8, 1971. The son of Walter A. and Harriot J. La Grone.

Education: Graduated from Del City High School, Del City, Oklahoma in May 1989; recieved a Bachelor of Science in Electrical Engineering from Oklahoma State University, Stillwater, Oklahoma, in May 1993. Completed the requirements for the Master of Science degree with a major in Electrical Engineering in May 1995.

Experience: Worked as a graduate research assistant from 1993 to the present.

Name: Marcus Jeanfreau la Grone Date of Degree: May, 1995

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: A PHOTO-THERMAL NONDESTRUCTIVE EVALUATION METHOD FOR CERAMICS WITH APPLICATIONS FOR OTHER MATERIALS

Pages in Study: 61 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Scope and Method of Study: The purpose of this experiment was to evaluate a tool that could be used for the nondestructive testing of ceramics. The system relies on the fact that heat waves will propagation as a function of the material, or lack of material, present. By using a chopped laser to probe a sample, it is possible to locally heat a section of a sample. By examining the emitted infrared radiation it is possible to inspect a material for defects both above and below the surface. By using a narrow band, highly sensitive filter, it is possible to exploit the standard black body spectral emission and gain better than a fourteenth power dependency, hence a highly sensitive system.

Findings and Conclusions: This experimental photo-thermal system was able to detect defects in both ceramic and steel products. By varying the chopping rate it was possible to control the depth that the system could "see" into a sample. The system was highly accurate and reproducible. Resolutions were limited only by the choice of wavelength for the probe beam and the focusing lens. This system represents an improvement over other photo-thermal and photo-acoustic systems.

ADVISOR'S APPROVAL: ______________________________

```
DIM pal&(255)
DIM info(100), phase(100)

SCREEN 13

FOR x = 0 TO 255
 LINE (x, 0)-(x, 100), x
NEXT x

p = 1
r = 63
FOR x = 0 TO 63
 g = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 1 TO 63
 r = 63 - x
 b = x / 2
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 b = 32 + x / 2
 g = 63 - x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 r = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x



DIM noise(30), raw(30)
DIM sortit(63, 2)
nm$ = STR$(nm)
OPEN "area" + MID$(nm$, 2, LEN(nm$) - 1) + ".iii" FOR OUTPUT AS #2

IBMBASE = &HD800
IBMBASE$ = "D800"
CONST data.out = &H3FC
CONST data.in = &H3FE
CONST data.avail = &H800
```

```
DEF SEG = IBMBASE      'SET BASE ADDRESS TO MATCH BOARD ADDRESS
GOSUB dumb

OPEN "com2:19200,n,8,2,cs,ds,cd" FOR RANDOM AS #1
PRINT #1, "    "
PRINT #1, "G"
INPUT #1, g: sr510gain = gain(g): PRINT #2, g
PRINT #1, "P": INPUT #1, phase: PRINT #2, phase
PRINT #1, "F": INPUT #1, freq: PRINT #2, freq
forward = .25
backward = -.2

xtarget = 0
ytarget = 0
xlocation = 0
ylocation = 0
walk = 0
speed = 0


CLS


GOSUB GoOn

xtarget = 0: ytarget = 0
'GOSUB GoAbsolute
 GOSUB dumb
  stuff$ = "SG50": GOSUB SendIt: GOSUB dumb: stuff$ = "SE255": GOSUB SendIt
 GOSUB dumb
 stuff$ = "SA8000000": GOSUB SendIt: GOSUB dumb

 stuff$ = "SD8000000": GOSUB SendIt: GOSUB dumb
 stuff$ = "SP1": GOSUB SendIt: GOSUB dumb

RMSvalue = 0: rms = 0
speed = 500: GOSUB GoSpeed
GOSUB HowFast
tm = TIMER + 1
walk = 0
peat:

 a$ = INKEY$: IF a$ = " " THEN walk = 0
 IF a$ = CHR$(27) THEN GOSUB GoHome
 IF a$ = "g" THEN GOSUB GoAbsolute

 IF a$ = "6" THEN xtarget = xtarget + 8.3333
 IF a$ = "4" THEN xtarget = xtarget - 8.3333
 IF a$ = "8" THEN ytarget = ytarget + 8.3333
 IF a$ = "2" THEN ytarget = ytarget - 8.3333

 IF a$ = "f" THEN speed = 2000: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "s" THEN speed = 500: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "F" THEN speed = 5000: GOSUB GoSpeed: GOSUB HowFast
```

```
 IF a$ = "S" THEN speed = 100: GOSUB GoSpeed: GOSUB HowFast
 IF a$ = "q" THEN END
 IF a$ = "?" THEN GOTO SnarfData
 IF TIMER > tm THEN tm = TIMER + 1: GOSUB Stats

GOTO peat

SR510v:
 PRINT #1, "Q"
 INPUT #1, volt
 v = volt / (1E-09)
 ap$ = "n": IF v > 1000 THEN v = v / 1000: ap$ = "æ"
 IF v > 1000 THEN v = v / 1000: ap$ = "m"
 v = INT(v * 1000 + .5) / 1000
 volt$ = STR$(v) + " " + ap$ + "V"
 RETURN



GETd812:
stuff = 0
dead1:
 IF (PEEK(data.avail) AND 2) = 0 THEN GOTO dead1
 stuff = PEEK(data.in)
 'PRINT stuff, CHR$(stuff)
RETURN

GoHome:
 stuff$ = "GH": GOSUB SendIt
RETURN

GoOn:
 stuff$ = "MN": GOSUB SendIt
RETURN

GoAbsolute:
 xtarget = CLNG(xtarget + .5)
 ytarget = CLNG(ytarget + .5)

 stuff$ = "3MA" + STR$(xtarget)
 GOSUB SendIt
 stuff$ = "4MA" + STR$(ytarget)
 GOSUB SendIt
RETURN

GoSpeed:
 GOSUB dumb
 stuff$ = "SV" + STR$(speed)
 GOSUB SendIt
RETURN

GoSpeedX:
 GOSUB dumb
 stuff$ = "3SV" + STR$(speed)
 GOSUB SendIt
```

```
RETURN

GoSpeedY:
 GOSUB dumb
 stuff$ = "4SV" + STR$(speed)
 GOSUB SendIt
RETURN


SendIt:
'PRINT "sendit:"; stuff$
 GOSUB dumb
 stuff$ = stuff$ + CHR$(13)
 count = LEN(stuff$)
 FOR s = 1 TO count
  a$ = MID$(stuff$, s, 1)
  stuff = ASC(a$)
retry:
  IF (PEEK(data.avail) AND 1) = 0 THEN
   POKE data.out, stuff
   POKE data.out + 3, stuff
   'PRINT "send "; CHR$(stuff), stuff
  ELSE
   GOSUB dumb
   GOTO retry
  END IF

 NEXT
RETURN

HowFast:
 GOSUB dumb
 stuff$ = "3TY": GOSUB SendIt
 a$ = ""
dead9:
  GOSUB GETd812
  IF stuff <> ASC("Y") THEN GOTO dead9
dead10:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead10
  speed = VAL(a$)
  GOSUB dumb
RETURN

WhereAmI:
  stuff$ = "3TP": GOSUB SendIt
  a$ = ""
dead3:
  GOSUB GETd812
  IF stuff <> ASC("P") THEN GOTO dead3
dead4:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead4
```

```
  xlocation = VAL(a$)
  GOSUB dumb
  stuff$ = "4TP": GOSUB SendIt
 a$ = ""
dead5:
  GOSUB GETd812
  IF stuff <> ASC("P") THEN GOTO dead5
dead6:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead6
  GOSUB dumb
  ylocation = VAL(a$)
RETURN

dumb:
dead0:
 IF (PEEK(data.avail) AND 2) <> 0 THEN a = PEEK(data.in): GOTO dead0
 ' PRINT a
RETURN

AreWeDone:
  stuff$ = "3TS": GOSUB SendIt
  a$ = ""
dead11:
  GOSUB GETd812
  IF stuff <> ASC("S") THEN GOTO dead11
dead12:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead12
  st1 = VAL(a$)
  GOSUB dumb
  stuff$ = "4TS": GOSUB SendIt
 a$ = ""
dead13:
  GOSUB GETd812
  IF stuff <> ASC("S") THEN GOTO dead13
dead14:
  GOSUB GETd812
  a$ = a$ + CHR$(stuff)
  IF stuff <> 13 THEN GOTO dead14
  GOSUB dumb
  st2 = VAL(a$)
  st3 = 0
  IF (st1 AND 1) AND (st2 AND 1) THEN st3 = 1
RETURN

Stats:
 GOSUB WhereAmI
 LOCATE 1, 1: PRINT USING "X-Loc: ####.###    Y-Loc: ####.###"; xlocation * .06; ylocation * .06
 PRINT USING "X-Tar: ####.###    Y-Tar: ####.###"; xtarget * .06; ytarget * .06
 PRINT USING "Speed: #####"; speed
 GOSUB SR510v
 PRINT "Voltage: " + volt$ + "        "
```

```
'GOSUB AreWeDone: PRINT USING "ST1 ###   ST2 ###  ST3 #"; st1; st2; st3


 rms = rms + 1
 raw(rms) = volt
 IF rms < 30 THEN GOTO gondi

 q = 0: FOR a = 1 TO 30: q = q + raw(a): NEXT a: q = q / 30
 FOR a = 1 TO 30: noise(a) = raw(a) - q: NEXT a: ref = q
 q = 0: FOR a = 1 TO 30: q = q + noise(a) * noise(a): NEXT

 q = SQR(q / 30)

 q = (q / ref) * 100
 LOCATE 11, 10: PRINT USING "RMS Noise:###.# %"; q
 rms = 29
 FOR a = 1 TO 29: raw(a) = raw(a + 1): NEXT a
gondi:
RETURN

SnarfData:
  CLS
  sttime = TIMER
  counter = 0
  PRINT "1"

   xtarget = -50: ytarget = 0
  GOSUB GoAbsolute
here:
  GOSUB WhereAmI
  IF xlocation < (-20) THEN GOTO here
    speed = 5000: GOSUB GoSpeedY
   speed = 5000: GOSUB GoSpeedX


  odd = 0

  vmax = 0: vmin = 4096
   CLS
  sttime = TIMER
  alt = 0

   speed = 15000: GOSUB GoSpeedX
   xtarget = -50: GOSUB GoAbsolute
check3:
   GOSUB WhereAmI
   IF ABS(xlocation - xtarget) > 30 THEN GOTO check3



   FOR ytag = -5000 TO 5000 STEP 100
   ytarget = CLNG(ytag / .06)
   GOSUB GoAbsolute
check2:
```

```
  GOSUB WhereAmI
  IF ABS(ylocation - ytarget) > 30 THEN GOTO check2



  speed = 45000: GOSUB GoSpeedX
  xtarget = 170000: tag = 1
  IF odd = 1 THEN xtarget = -500
  GOSUB GoAbsolute
check:

GOSUB WhereAmI

  volt = 0: phs = 0
  FOR vcount = 1 TO tft: CALL LasPow: volt = volt + 1024 * tlaspow
  GOSUB atan2
  phs = phs + angle
  NEXT: volt = CLNG(volt / tft): phs = CLNG(phs / tft * 10)

  IF volt > vmax THEN vmax = volt
  IF volt < vmin THEN vmin = volt
   xx = CLNG(((xlocation * .06)) / 100)
   yy = CLNG(((ylocation * .06) + 5000) / 100)
   xxx = xx * 2
   yyy = yy * 2
   quit$ = INKEY$: IF quit$ = CHR$(27) THEN GOTO OldMan
    IF xx > 0 AND xx < 100 THEN
      info(xx) = volt
      phase(xx) = phs
      LINE (xxx + 1, yyy + 1)-(xxx, yyy), 256 - volt, B
      IF volt > vmax THEN vmax = volt
      IF volt < vmin THEN vmin = volt
    END IF
   IF (odd = 0) AND (xlocation < 167000) THEN GOTO check
   IF (odd = 1) AND (xlocation > 0) THEN GOTO check

  speed = 150000: GOSUB GoSpeed
  odd = odd + 1: IF odd > 1 THEN odd = 0
  FOR we = 1 TO 100
   PRINT #2, info(we): info(we) = 0
   PRINT #2, phase(we): phase(we) = 0
   NEXT

  ytarget = ytarget - 150
  GOSUB GoAbsolute

check6:
  GOSUB WhereAmI
  IF SQR((ylocation - ytarget) ^ 2) > 20 THEN GOTO check6



   NEXT ytag
```

```
OldMan:
   PRINT #2, 0, 0, -999
   PRINT #2, vmax, vmiv
CLOSE 2
CLOSE 1
 PRINT TIMER - sttime
 PRINT vmax, vmin
GOSUB GoHome:
a$ = INPUT$(2)
END

atan2:
 ap = ABS(laspow1)
 bp = ABS(laspow2)
 a = laspow1
 b = laspow2
 IF (a = 0) AND (b = 0) THEN RETURN
 power = SQR(a ^ 2 + b ^ 2)

 IF a > 0 AND b > 0 THEN q = 0
 IF a < 0 AND b > 0 THEN q = 1
 IF a < 0 AND b < 0 THEN q = 2
 IF a > 0 AND b < 0 THEN q = 3

 IF bp >= ap THEN
    eep = 0
    angle = 90 - rtd * ATN(a / b)
     IF q > 1 THEN angle = angle + 180

   END IF
 IF ap > bp THEN
     eep = 1
     angle = rtd * ATN(b / a)
     IF q > 0 THEN angle = angle + 180
     IF q = 3 THEN angle = angle + 180
   END IF

RETURN

SUB LasPow
SHARED addr, laspow1, laspow2, tlaspow
 s = INP(addr)
 OUT addr, 0
 OUT addr + 3, 0
piddy:
 s = INP(addr)
 IF (s AND 1) = 0 THEN GOTO piddy
 lpow = INP(addr + &HA)
 hpow = INP(addr + &HA)
 laspow1 = (hpow * 256 + lpow - 2048)

 s = INP(addr)
 OUT addr, 1
 OUT addr + 3, 0
piddy2:
```

```
  s = INP(addr)
  IF (s AND 1) = 0 THEN GOTO piddy2
  lpow = INP(addr + &HA)
  hpow = INP(addr + &HA)
  laspow2 = (hpow * 256 + lpow - 2048)

tlaspow = ((laspow1 ^ 2 + laspow2 ^ 2) ^ .5) / 4096
END SUB
```

Appendix B.

This software was written in QBasic and compiled. It takes the ".iii" files generated from the scaning program and compresses them, and stamps them with a version number, time/date, gain, phase and frequency information (pertaining to the settings on the SR-510 lock-in amplifier).

```
CLS
INPUT "Sample:"; nm$
DIM q(100, 100) AS INTEGER
DIM w(100, 100) AS INTEGER
vmx = -10000: vmn = 10000


OPEN "area" + MID$(nm$, 1, LEN(nm$)) + ".iii" FOR INPUT AS #1
OPEN "area" + MID$(nm$, 1, LEN(nm$)) + ".wlf" FOR OUTPUT AS #2

 INPUT #1, gain
 INPUT #1, phase
 INPUT #1, freq

FOR y = 1 TO 100
 FOR x = 1 TO 100
 INPUT #1, q(x, y)
 INPUT #1, w(x, y)
 IF q(x, y) > vmx THEN vmx = q(x, y)
 IF q(x, y) < vmn THEN vmn = q(x, y)
 NEXT
NEXT



 PRINT #2, "WOLF"
 PRINT #2, 3.1
 PRINT #2, DATE$
 PRINT #2, gain
 PRINT #2, phase
 PRINT #2, freq
 PRINT #2, vmx
 PRINT #2, vmn

FOR y = 1 TO 100
FOR x = 1 TO 100
    v = q(x, y) + 2048
    hs = INT(v / 256)
    ls = v MOD 256
```

```
      PRINT #2, CHR$(hs);
      PRINT #2, CHR$(ls);

    v = w(x, y)
    hs = INT(v / 256)
    ls = v MOD 256

      PRINT #2, CHR$(hs);
      PRINT #2, CHR$(ls);

NEXT: NEXT
```

Apendix C.

This software was written in QBasic. It was not compiled due to the fact that the supplied compiler could not deal with large memory models. This program will load up to three different images. It can display images side by side, or in a diferential form. Indvidual images may be displayed full screen with gradiated index on the side. This program expects to load ".wlf" files (Appendix B covers the ".iii" to ".wlf" software, Apendix D. discusses how these files are encoded)

```
DEF fncv (r, g, b) = INT(r) + 256 * INT(g) + 65536 * INT(b)
DIM pal&(255)
DIM a(100, 100, 2) AS INTEGER
DIM files$(2)
SCREEN 13
CLS
FOR x = 0 TO 255
 LINE (x, 0)-(x, 10), x
NEXT x

p = 1
r = 63
FOR x = 0 TO 63
 g = x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 1 TO 63
 r = 63 - x
 b = x / 2
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 b = 32 + x / 2
 g = 63 - x
 pal&(p) = fncv(r, g, b)
 PALETTE p, pal&(p)
 p = p + 1
NEXT x

FOR x = 0 TO 63
 r = x
 pal&(p) = fncv(r, g, b)
```

```
 PALETTE p, pal&(p)
 p = p + 1
NEXT x
p = 0: r = 63: g = 63: b = 63: pal&(p) = fncv(r, g, b): PALETTE p, pal&(p)


menuarea:
 rover = 1
 nf = 0
 dif = 0
peatmenu:
CLS
COLOR 200
 PRINT "1) "; : IF nf = 0 THEN COLOR 20: PRINT "<none>" ELSE PRINT file$(0)
COLOR 200
 PRINT "2) "; : IF nf < 2 THEN COLOR 20: PRINT "<none>" ELSE PRINT file$(1)
 PRINT : PRINT
COLOR 200
 PRINT "3) ";
 IF nf <> 2 THEN dif = 0
 IF (dif = 0) AND (nf < 3) THEN COLOR 20: PRINT "<none>"
 IF dif = 0 AND nf = 3 THEN PRINT file$(2)
 IF (dif = 1) AND nf = 2 THEN COLOR 120: PRINT " #1 vs #2 "
COLOR 200


 PRINT : PRINT :
 PRINT "A)dd File"
 PRINT "R)emove"
 PRINT "D)ifference"
 PRINT "P)lot All"
 PRINT "E)xpanded: "; : IF rover = 2 THEN PRINT "ON" ELSE PRINT "OFF"
 PRINT "<ESC> Reset"
 PRINT "Q)uit"


 i$ = INPUT$(1)
 IF (i$ = "d" OR i$ = "D") AND (nf = 2) THEN
        dif = dif + 1
        IF dif = 1 THEN GOSUB MakeDif
       END IF
 IF dif > 1 THEN dif = 0
 IF (i$ = "a" OR i$ = "A") AND (nf < 3) THEN
            CLS
            INPUT "Name:"; file$(nf)
            n = nf: nm$ = file$(n)
            GOSUB GetIt
            nf = nf + 1
          END IF
 IF i$ = "r" OR i$ = "R" THEN nf = nf - 1
 IF (i$ = "p" OR i$ = "P") AND (nf > 0) THEN
           CLS
           FOR q = 1 TO nf
             n = q - 1
            GOSUB PlotIt
```

```
            NEXT
            IF dif = 1 THEN n = 2: GOSUB PlotIt
            i$ = INPUT$(1)
          END IF
   IF i$ = "q" OR i$ = "Q" THEN END
   IF i$ = CHR$(27) THEN GOTO menuarea
   IF i$ = "e" OR i$ = "E" THEN rover = rover + 1: IF rover > 2 THEN rover = 1
   i = VAL(i$)
   IF i > 0 AND (nf + dif) >= i THEN
      CLS
      n = i - 1
      GOSUB PlotItBig:
      i$ = INPUT$(1)
      IF i$ = CHR$(13) THEN GOSUB Off2Disk
      END IF




   GOTO peatmenu

  'nm$ = "area95": n = 0: GOSUB GetIt: GOSUB PlotIt
  'nm$ = "area97": n = 1: GOSUB GetIt: GOSUB PlotIt
  'GOSUB PlotDif


  END
  MakeDif:
  vmn(2) = 10000: vmx(2) = -10000
  rng0 = vmx(0) - vmn(0): rng1 = vmx(1) - vmn(1): scl = rng0 / rng1
   file$(2) = "#1 vs #2"
   vers(2) = 0
   FOR y = 1 TO 100
   FOR x = 1 TO 100
    v1 = a(x, y, 1) * scl: v0 = a(x, y, 0)
    d = v1: IF v0 < v1 THEN d = v0
    IF d = 0 THEN d = 1
    v = (ABS((v1 - v0) / d) * 196) MOD 254
    IF v > vmx(2) THEN vmx(2) = v
    IF v < vmn(2) THEN vmn(2) = v
    a(x, y, 2) = v
   NEXT
   NEXT
  RETURN

  PlotIt:
  COLOR 254
  LOCATE 1, 1 + n * 13: PRINT file$(n);

  IF vers(n) > 0 THEN
   LOCATE 20, 1 + n * 13: PRINT d$(n);
   IF vers(n) > 1 THEN
      LOCATE 23, 1 + n * 13: PRINT "Ph :"; phase(n);
      LOCATE 24, 1 + n * 13: PRINT "Frq:"; freq(n);
```

```
  END IF
END IF
 stretch = rover * 254 / (vmx(n) - vmn(n))
  FOR y = 1 TO 100
  FOR x = 1 TO 100
   v = a(x, y, n) - vmn(n)
   IF vers(n) < 0 THEN v = v * .25 + 128
   vv = 256 - v * stretch: IF vv > 254 THEN vv = 254
   IF vv < 1 THEN vv = 1
   PSET (x + 100 * n, y + 10), vv
  NEXT
  NEXT
RETURN

PlotItBig:
COLOR 254
 stretch = rover * 254 / (vmx(n) - vmn(n))
  FOR y = 1 TO 100
  FOR x = 1 TO 100
   v = a(x, y, n) - vmn(n)
   IF vers(n) < 0 THEN v = v * .25 + 128
   vv = 256 - v * stretch: IF vv > 254 THEN vv = 254
   IF vv < 1 THEN vv = 1

   yy = y * 2 - 2
   xx = x * 2.5 - 2
   LINE (xx, yy)-(xx + 3, yy + 2), vv, BF
  NEXT
  NEXT
  FOR y = 1 TO 254: LINE (253, y * .78)-(263, y * .78), y: NEXT
  FOR sc = 0 TO 1 STEP .25
   LOCATE 24 * (1 - sc) + 1, 35
   COLOR 255 - 254 * sc
   PRINT sc;
  NEXT
RETURN

GetIt:
OPEN nm$ + ".wlf" FOR BINARY AS 1
T$ = INPUT$(6, 1)
GOSUB whip: vers(n) = z
i$ = INPUT$(1, 1)
d$(n) = INPUT$(11, 1)
GOSUB whip: g = z
IF vers(n) > 1 THEN
    GOSUB whip: phase(n) = z
    GOSUB whip: freq(n) = z
   END IF
GOSUB whip: vmx(n) = z
GOSUB whip: vmn(n) = z
'PRINT vmn, vmx, g, vers, d$: STOP
i$ = INPUT$(1, 1)
FOR y = 1 TO 100
   FOR x = 1 TO 100
   hs = ASC(INPUT$(1, 1))
```

```
  ls = ASC(INPUT$(1, 1))
   v = hs * 256 - 2048 + ls
   a(x, y, n) = v
 IF vers(n) >= 3 THEN
      hs = ASC(INPUT$(1, 1))
      ls = ASC(INPUT$(1, 1))
      v = hs * 256 + ls
      IF v > 180 THEN v = v - 360
      IF vers(n) > 3 THEN v = v / 10
      IF nf < 2 THEN a(x, y, n + 1) = v
    END IF
  NEXT
NEXT
CLOSE 1
v = 0: FOR y = 10 TO 90: v = v + a(y, 1, n): NEXT: vmn(n) = v / 80
v = 0: FOR y = 10 TO 90: v = v + a(y, 99, n): NEXT: vmx(n) = v / 80

IF vers(n) >= 3 AND nf < 2 THEN
     nf = nf + 1
     file$(nf) = file$(n) + " phs"
     vers(nf) = -1
     vmx(nf) = 180
     vmn(nf) = -180
    END IF
RETURN
whip:
 x$ = ""
ick:
  z$ = INPUT$(1, 1): IF z$ = CHR$(13) THEN RETURN
  IF z$ = "-" THEN x$ = "-"
  IF z$ = "." THEN x$ = x$ + "."
  IF (VAL(z$) = 0 AND z$ <> "0") THEN GOTO ick
  x$ = x$ + z$
  z = VAL(x$)
  GOTO ick

Off2Disk:
OPEN file$(n) + ".red" FOR OUTPUT AS #1
OPEN file$(n) + ".grn" FOR OUTPUT AS #2
OPEN file$(n) + ".blu" FOR OUTPUT AS #3
FOR y = 0 TO 199
FOR x = 0 TO 319
 c = POINT(x, y): p = pal&(c)
 b = 4 * INT(p / 65536): p = p MOD 65536
 g = 4 * INT(p / 256): r = 4 * (p MOD 256)
PRINT #1, CHR$(r);
PRINT #2, CHR$(g);
PRINT #3, CHR$(b);
NEXT
NEXT
CLOSE 3, 2, 1
RETURN
```

Appendix D.

".WLF" Encoding Scheme

The filename is expected to be "<NAME>.WLF".

HEADER FORMAT

Field	ID	Format	Size	Versions
header	WOLF<CRLF>	ASCII	6 bytes	1.0+
version	#.#<CRLF>	ASCII	3+ bytes	1.0+
date	<as per DOS>	ASCII	11 bytes	1.0+
gain	#(1-24)<CRLF>	ASCII	3+ bytes	1.0+
phase	#.#<CRLF>	ASCII	3+ bytes	2.0+
frequency	#.#<CRLF>	ASCII	3+ bytes	2.0+
max value	#<CRLF>	ASCII	3+ bytes	1.0+
min value	#<CRLF>	ASCII	3+ bytes	1.0+

DATA FORMAT

versions 1-3.5 are of a 100 by 100 array

(all versions) VALUE<High Byte><Low Byte>

(version 3.0+) PHASE<High Byte><Low Byte>

repeat until end.

for example a version 3.0 data file would be as:

<value1.1><phase1.1><value1.2><phase1.2> ... <value100.100><phase100.100>

VITA

Marcus Jeanfreau la Grone

Candidate for the Degree of

Master of Science

Thesis: A PHOTO-THERMAL NON-DESTRUCTIVE EVALUATION METHOD FOR CERAMICS WITH APPLICATIONS FOR OTHER MATERIALS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born at Grissom Air Force Base, Indiana, on October 8, 1971. The son of Walter A. and Harriot J. La Grone.

Education: Graduated from Del City High School, Del City, Oklahoma in May 1989; recieved a Bachelor of Science in Electrical Engineering from Oklahoma State University, Stillwater, Oklahoma, in May 1993. Completed the requirements for the Master of Science degree with a major in Electrical Engineering in May 1995.

Experience: Worked as a graduate research assistant from 1993 to the present.