Infrared Thermography Flight Experimentation

NASA GRANT: NCC4-156

FINAL REPORT

By:
Robert C. Blanchard
Matthew L. Carter
Michael F. Kirsch

George Washington University
NASA Langley Research Center
Mail Stop 335
Hampton, VA 23681

Abstract

Analysis was done on IR data collected by DFRC on May 8, 2002. This includes the generation of a movie to initially examine the IR flight data. The production of the movie was challenged by the volume of data that needed to be processed, namely 40,500 images with each image (256 x 252) containing over 264 million points (pixel depth 4096). It was also observed during the initial analysis that the RTD surface coating has a different emissivity than the surroundings. This fact added unexpected complexity in obtaining a correlation between RTD data and IR data. A scheme was devised to generate IR data near the RTD location which is not affected by the surface coating This scheme is valid as long as the surface temperature as measured does not change too much over a few pixel distances from the RTD location. After obtaining IR data near the RTD location, it is possible to make a direct comparison with the temperature as measured during the flight after adjusting for the camera's auto scaling. The IR data seems to correlate well to the flight temperature data at three of the four RTD locations. The maximum count intensity occurs closely to the maximum temperature as measured during flight. At one location (RTD #3), there is poor correlation and this must be investigated before any further progress is possible. However, with successful comparisons at three locations, it seems there is great potential to be able to find a calibration curve for the data. Moreover, as such it will be possible to measure temperature directly from the IR data in the near future.

List of Figures and Tables

Figure Page

Table

Nomenclature

CFD:	Computational Fluid Dynamics
DFRC:	Dryden Flight Research Center
IR:	Infrared
RTD:	Resistive Temperature Device

Introduction

Infrared (IR) thermography is a developing noninvasive discipline for gathering data on flow phenomena. IR cameras sensitive in the infrared bands to surface emissivity, which is directly related to surface temperature, have been utilized in wind tunnels and during flight experiments at subsonic to hypersonic conditions. Several advantages exist which make this discipline attractive for experimentation (such as, its global nature, and not requiring modification of the target, to name a few). Flight data has been collected for a variety of vehicles under three basic camera-target configurations. That is, (1) with the camera and the target on one vehicle[1], (2) with the camera and the target on different vehicles[2], and (3) during remote ground installations wherein the target image is captured and automatically tracked on a ground fixed (but mobile) pedestal[3]. Recent infrared data were obtained under arrangement (1) using a low-sweep flight test article mounted on the center pylon of the NASA Dryden Flight Research Center (DFRC) F-15B aircraft as depicted in Fig. 1-1, left photo. To date, the data show that extended runs of natural laminar flow are feasible and maintainable at supersonic conditions[4].

In-flight flow experimentation using infrared imaging is a valuable tool to analyze the flow and temperature characteristics of flight vehicles. In particular, mapping of the infrared images into global temperatures and validation of this technique would provide order of magnitude improvements to the infrared thermography process. The purpose of the effort discussed in this report is to support the continuing supersonic laminar flow flight experiments, through analyses of the flight infrared data, by developing the technology required to generate global surface temperature images, and validating these with onboard thermocouples (or equivalent) sensors and numerical CFD solutions. This effort supports DFRC activities to further develop and use infrared technology in laminar flow flight experiments.

The specific goal of this research is to further develop and test a method of extracting temperature information from infrared images using the IR imaging data recently collected on the F-15B. Flight data for this research was collected at Dryden Flight Research Center (DFRC) on May 8, 2002. The test article used is a supersonic natural laminar flow test article (fin) mounted to the fuselage, (on the centerline drop tank location); see Fig. 1-1, right photo. There are four RTDs, mounted on the surface of the fin in view of the camera, which are relevant to this research. Developing temperature information from the collection of flight IR imaging data involves exploring the IR flight data and the accompanying ancillary data in conjunction with the RTD data. RTD data will be used to compare and possibly scale the infrared temperature data. Clearly, once this temperature extraction process is developed and tested, data from the RTDs will no longer be necessary since the temperature will be available from the IR camera image data. Ultimate use of this research is to have the technology to collect precise global temperature data on the surface as the flow transitions from laminar to turbulent.

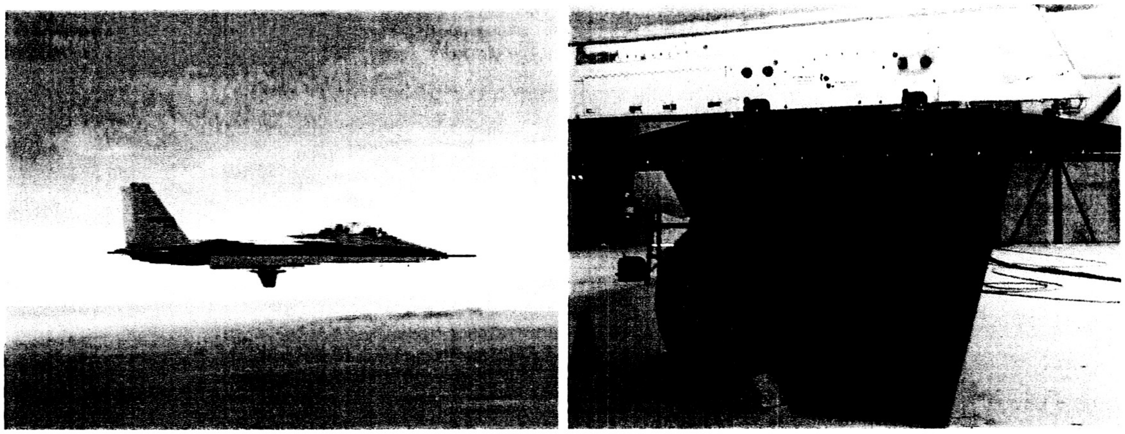

Figure 1: IR Camera and Test article mounted to F-15B aircraft

Flight Conditions

Shown on Fig. 2 is the altitude and Mach number flight profile flown by the F-15B aircraft during the conduct of the laminar-flow experiment. The times shown in the profile are the approximate times during which the IR flight data were collected.

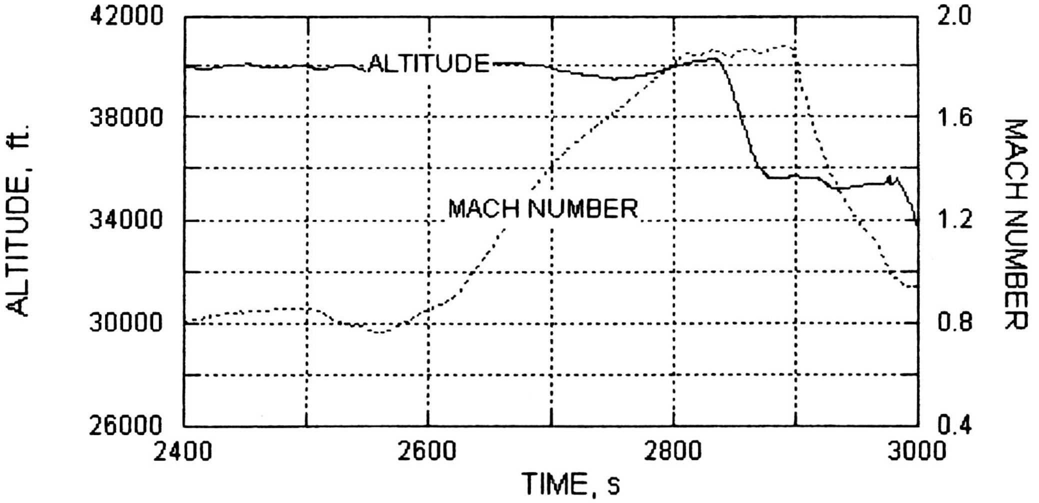

Fig. 2: Altitude and Mach number flight profile of the F15B aircraft during data acquisition.

Data Analyses

Data Acquisition

The laminar flow experiment IR camera and associated flight recording equipment generated approximately 10 minutes of IR flight data. Specifically, during flight, data is recorded at 60 frames per second in binary format. After the flight, the flight data is stripped from the flight data recorder and placed on DVD_RAM discs. The data on the discs consists of 45 files ("IR01" – "IR45") where each file contains 900 frames. The file format contains a header followed by the flight data, i.e. an IR image of the fin in terms of pixels. Each image consists of 256 by 252 matrix with 4096 bit pixel depth, or counts. Information from the 4KB header is saved separately and later displayed with the IR image. All 45 files are read sequentially in binary format by the algorithm, discussed in the Appendix.

Initial Analyses

The first task after receiving the flight data is to devise a scheme to visually display the image data in order to assess the data content. It was decided after randomly viewing selected images of the data that the optimum process of viewing the entire data set is to generate a movie. This task was accomplished in a series of steps that now have been incorporated into a single MATLAB (v.6.5) algorithm, see appendix.

An outline of the process to visually display the flight data is as follows. Each of the 900 image matrix files (40,500 images in all) is separated from the header data so that information from the header can be displayed on a figure window along with the data images. The specific header information displayed from each data frame is the time of the image frame along with the frame number. The data image is, thus, sequentially captured and saved as frames to be used to construct a movie. Finally, a movie is generated using existing MATLAB routines and by applying the appropriate compression-decompression (codec) codes. Several standard industry codecs were investigated for best compression or smallest file size while retaining image content. The codec chosen is Cinepak (by Radius)

The movie provided a great deal of insight into the character of the data set. Some problems were observed however, such as a "flicker". The flicker occurred mainly in the color bar while viewing the movie. The flicker as observed is a change in the relative color associated with a given pixel count. This can be accounted for by the fact that the color bar is directly scaled for each individual frame. This means that whatever is the highest pixel count in the frame becomes the max on the color bar (256) and the lowest count is associated with the min (0). After max and min are set, MATLAB linearly interpolates between them, and therefore if two frames have different max and min pixel counts the scales of the color bars will be different. This is the root cause of the flicker and will be dealt with in the future. Another observation derived from the movie is that there appeared a "discontinuity" in the associated fin colors interspersed at seemingly random times during the movie. This "discontinuity" has the appearance of a change in the airflow. Several unsuccessful attempts were undertaken to try to comprehend this, but without success due to a lack of understanding of the cause. During later analysis the cause of this discontinuity was determined to be camera rescaling as discussed later in the report.

RTD Data

RTD data is simultaneously collected with the IR data during flight. These data are extremely important for the infrared calibration and verification process. Figure 3 shows a scaled sketch of the test fin, dimensions are given in inches, along with the four RTDs and their locations with respect to each other. The airflow for the data in this report is from right to left, as illustrated in the figure. The terminology used in this report is; RTD #1 located in the upper right corner and is called "top-right", RTD #2 is the lower right and is called "bottom-right", RTD #3 is located in the upper left and is called ""top-left" and RTD #4 is located in the lower left and is referred to as "bottom-left".

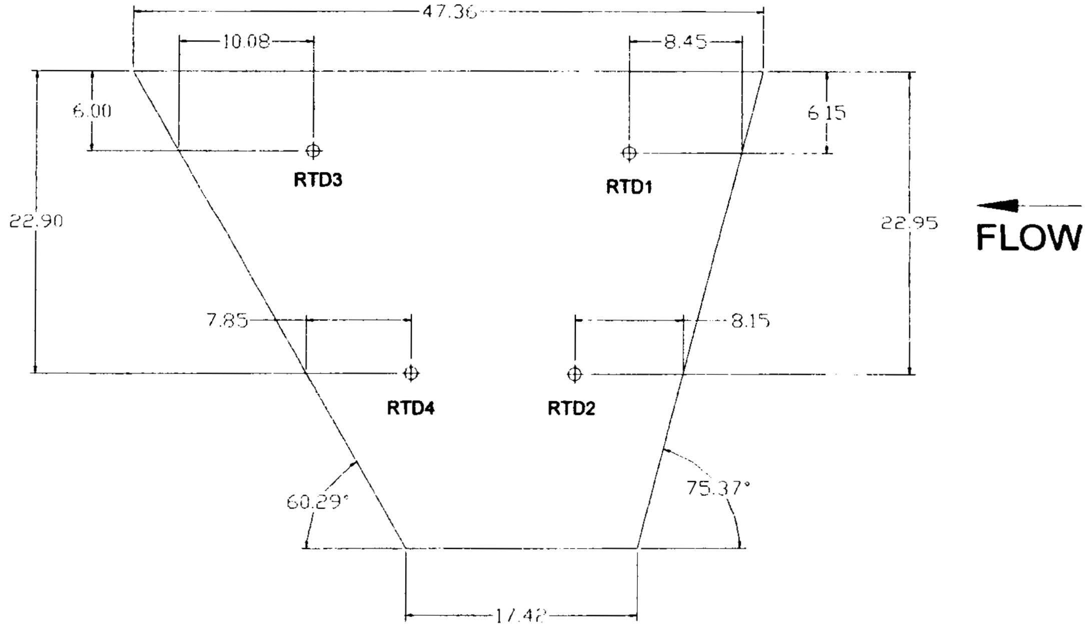

Fig. 3: Location and identification of the RTD sensors (facing the IR camera).

Fig. 4 shows the temperature data from the four RTDs taken during flight. The time period covered is the same as the infrared data. Also shown on the figure is the estimate of free-stream temperature during the same time interval. It is seen that the RTD #1 and #2, both near the leading edge of the fin, follow closely throughout the flight. However, the trailing edge sensors appear to start the flight with a difference of about 40 °F. RTD #4, located near the bottom edge of the test article, shows an initial temperature about 50 °F warmer than free-stream. Whereas, RTD #3, located near the top, is only about 10 °F warmer than free-stream which appears unusual relative to the other initial sensor values.

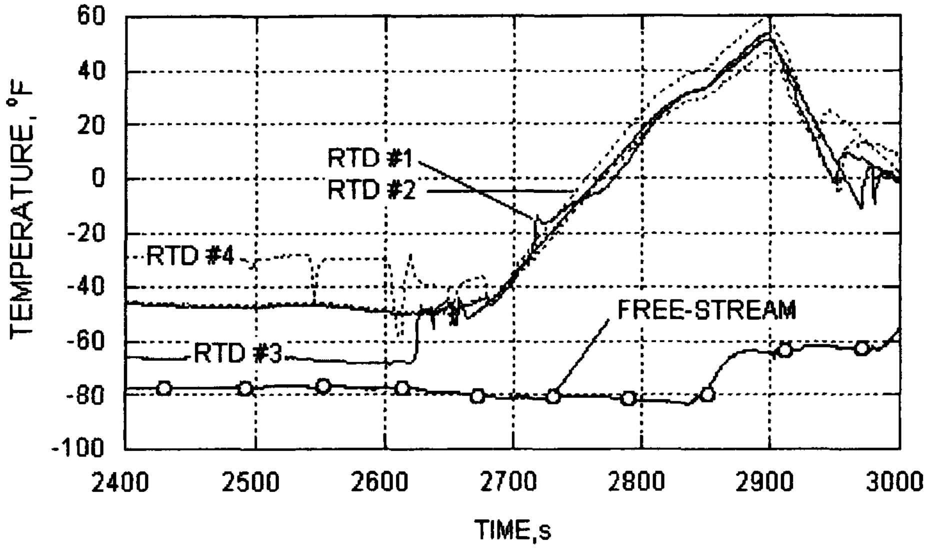

Fig. 4: RTD flight data along with free-stream temperature.

Infrared Data

A necessary part of the analysis to transform the infrared data into temperature requires a comparison to the RTD data. Thus, it was necessary to capture data points from the IR data files at the RTD locations. Figure 5 shows the averaged infrared data at the four RTD locations. The averaging process will be discussed later.

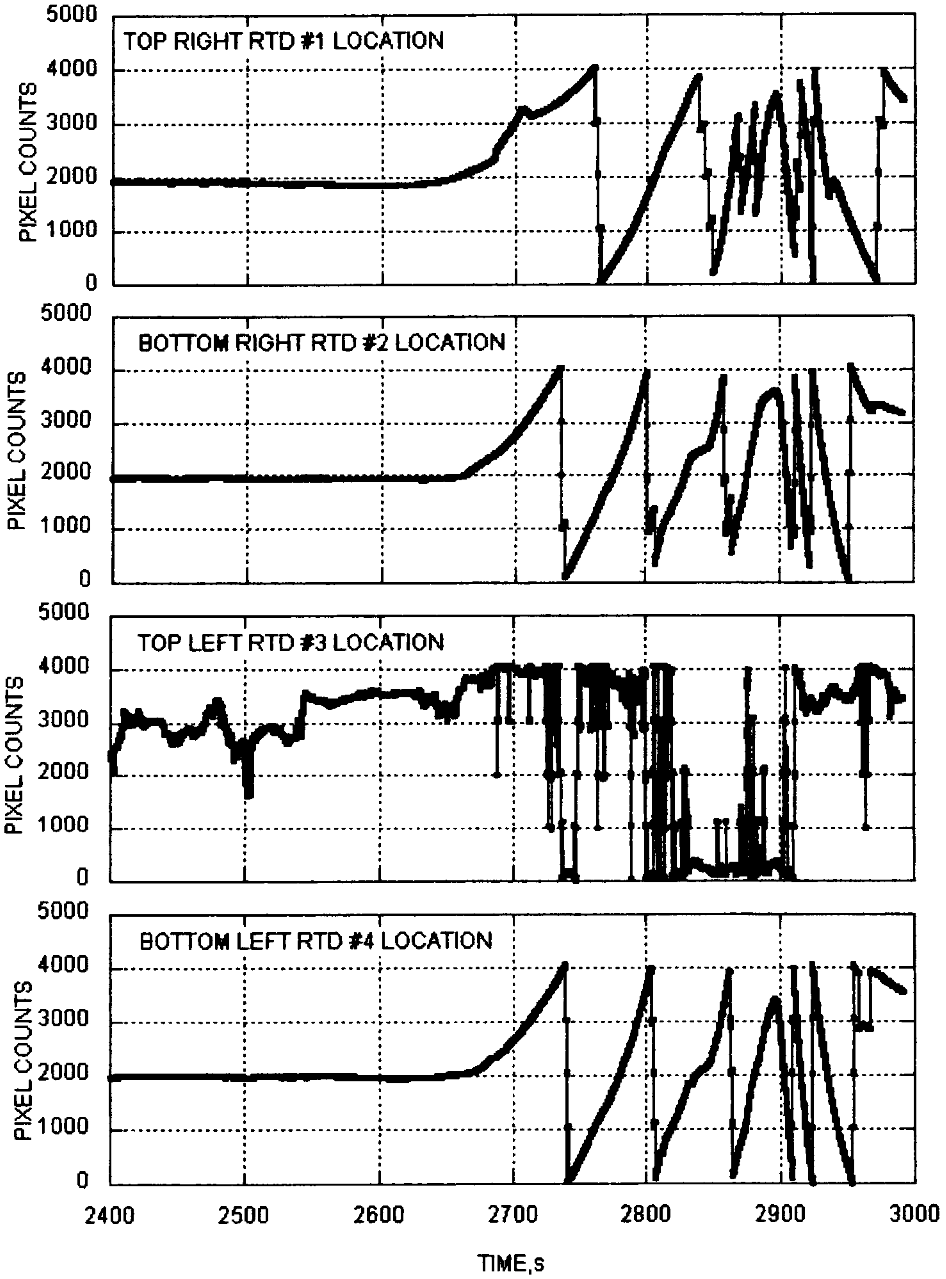

Fig. 5: Infrared pixel count data at each of the four RTD locations.

There are several features in Fig. 5 worth mentioning. First, it is clear that about three “up ranges” and three “down ranges” in the camera system took place during the data collection. Without extensive research into the camera's detail operations, it is postulated that the up ranging of the camera is triggered by a certain percentage of the image area (256 x 252 pixels) obtaining the 4096 maximum count range. Similarly, down ranging occurs when most of the pixels counts in a given frame are near zero. Since the graphs in Fig. 5 are for only a specific location, it is not necessary that at each specific location the maximum count range be obtained. In addition, it appears that near the RTD #3 location, the temperature is hotter than the scale of the IR camera can maintain. Thus, the counts are always near the maximum 4096 values throughout the data interval at this location. There is another point seen in the graph that needs further investigation. Namely, the times at which the camera up-ranged are not the same. These times should be the same at all locations if the earlier postulation on rescaling is correct (there is only one camera). As seen in Fig. 5 with the time-synchronized graphs, this is not happening and thus, the camera's auto scaling characteristics require further study.

Fig. 6 shows the infrared data after adjusting for the range changes at the four RTD locations. These adjustments for the range change provide certain advantages. First, it should eliminate the discontinuity that was observed in the initial analysis of the movie. This will present a more uniform scale distribution to the color map thereby proving a more meaningful correlation of IR intensity with time. Finally, it provides a way to make a direct measure with the temperature provided by the RTDs.

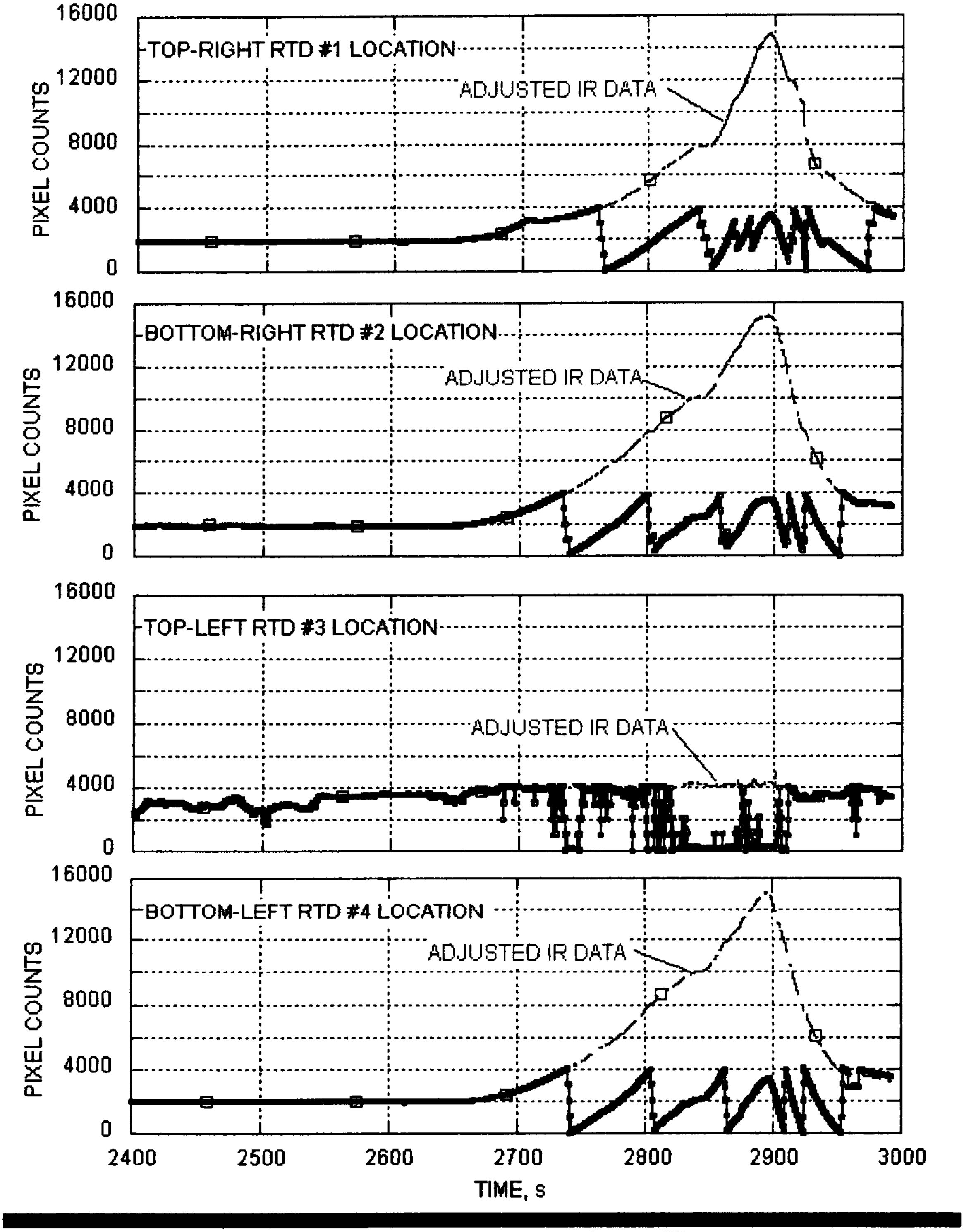

Fig. 6: Adjusted Infrared data at the four RTD locations

Comparison Analysis

It is necessary to make a comparison to the surface temperature as measured at the four RTD locations with the values obtained from the IR image. There is a problem to be resolved, however. The problem is that it was observed during the initial analysis of the infrared images that the emissivity in the vicinity of the RTD locations is different from the surrounding test article. An example of this is given in Fig. 7. Clearly on the image obtained at 209.2359 s (taken in the hangar, prior to flight), the four RTD locations are easily distinguished from the remaining test article. Discussion with DFRC indicated that there is a coating applied in the area of the RTDs. This paint changes the surface emissivity and thereby accounts for the difference observed. Since the object of the IR to temperature transformation is to apply the method to the entire surface where a different emissivity than the paint is evident, a way is needed to eliminate the effect of the paint in the subsequent analysis. To resolve the problem a technique has been developed. This technique involves moving a fixed number of pixels from the nominal RTD location and using the IR count value, or an average, at that location. Obviously, this approximate method relies on the fact that the temperature reported by the RTD should not change appreciably a few pixels away.

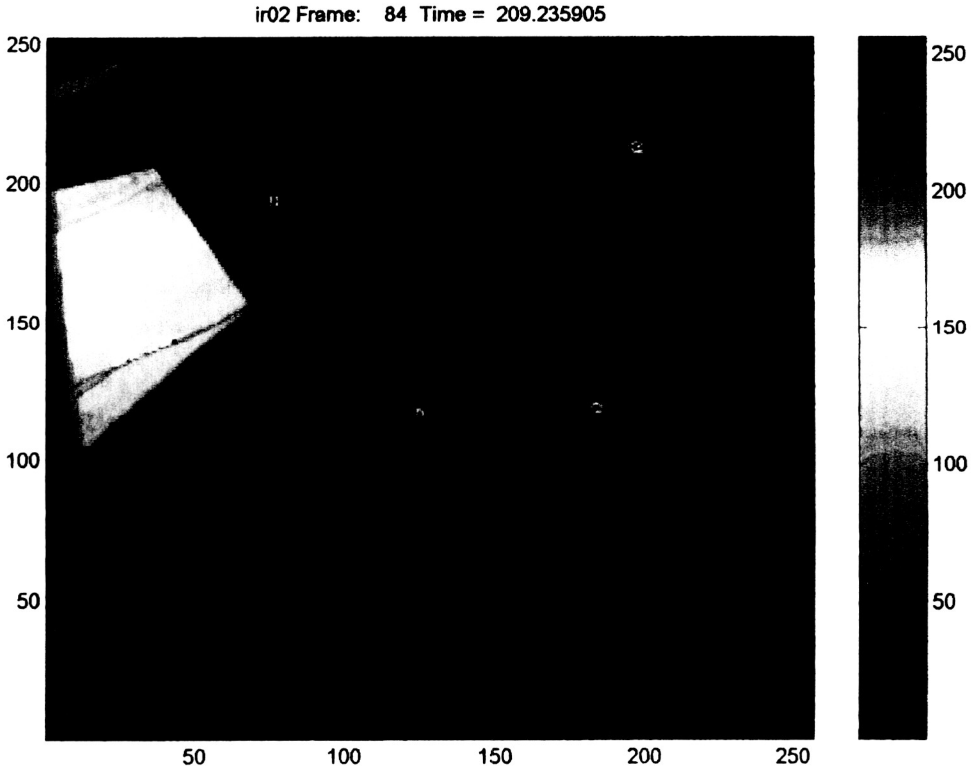

Fig. 7: Infrared test image taken prior to flight. (IR data scaled from 4096 to 256)

RTD Locations

In order to compare infrared to RTD data, the locations of each RTD had to be determined in the IR image data. The dimensions of the fin along with the location of the RTDs on the fin are given in Fig.3. This data is used to estimate a scale of pixel counts to inches. The scale was applied and the nominal locations of the RTDs as found in the IR image data. The following table, Table 1, shows the nominal RTD location in term of image pixels after iterations to get the location to within a pixel. The origin of the coordinate system for (x, y) is bottom left of the image, as seen in Fig. 7.

Table 1: Image Location of RTD

RTD	x (pixels)	y (pixels)
Top-Right (RTD #1)	197	212
Bottom Right (RTD #2)	183	118
Top Left (RTD #3)	76	194
Bottom Left (RTD #4)	124	117

Fig. 8 shows a zoomed section of the IR image data (Fig. 7) in the vicinity of the top right RTD. The image was obtained by dividing the raw data by 16 in order to correlate to a 256-color scale (shown to the right of the image). This color scale is relative for each image (i.e., the 256 scale is adjusted to the minimum and maximum of the count values of a given image, as mentioned earlier). Each square in the image represents one pixel location with the labeling at the center of the square. As seen on Fig. 8, there is an area about the nominal top-right RTD location (197,212) that corresponds to the painted area discussed earlier. The area is about 6 pixels square in size approximately centered about the nominal RTD location. To eliminate using the value of the IR data at the nominal RTD location one simply can move four pixels in any of the four directions and use the IR data at any of these locations. However, to do this examination for all the images and each of the four locations is impractical. In addition, using a fixed value is not workable since the size of the area changes as a function of time due to emissivity changes during flight creating different size areas. To get around this problem a scheme was devised. This scheme consisted of moving one pixel at a time and calculating the average image count about the nominal center (this produces a relatively high count value initially due to the high emittance of the paint). The average pixel count can be monitored as a function of distance from the nominal RTD location to find when there is little significant change in the average pixel count. In effect, this automatically finds a radius about the nominal RTD location whereby the painted surface can be avoided when generating a comparison to the RTD temperatures later. Once the radius is determined, then the average value of the IR count data at the nominal location minus the radius can be used. After analysis, it was determined that a fixed radius would be easier to implement and a constant distance from the nominal RTD centers for all images has its advantages for later interpretation.

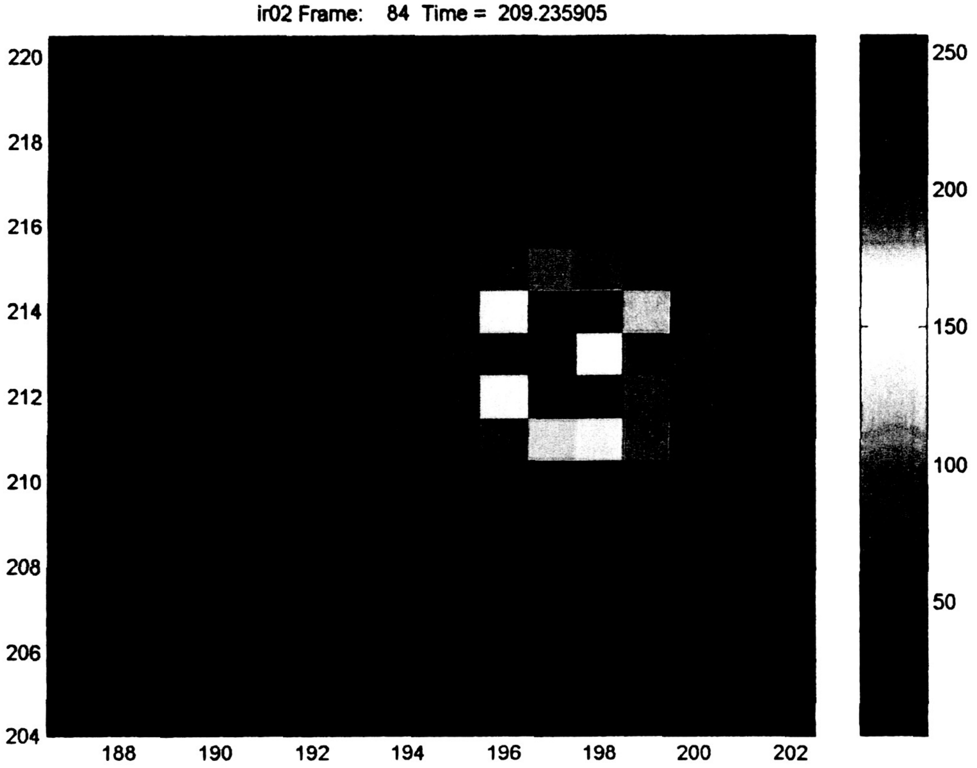

Fig. 8: Zoomed IR image data near RTD #1 (count data divided by 16 to color-scale)

Fig. 9 shows the results of an averaging scheme for the IR image data shown in Fig. 8. The pixel location (197, 212), corresponds to zero distance on the graph. It is seen that the desired background IR value in the vicinity of the RTD is achieved at approximately 8 to 10 pixels away from the nominal center. The approximate resolution of the camera is 3/16 in. per pixel, resulting in a physical distance of about 2 in. It is necessary to assume that the temperature does not change appreciably over this distance to make this scheme viable.

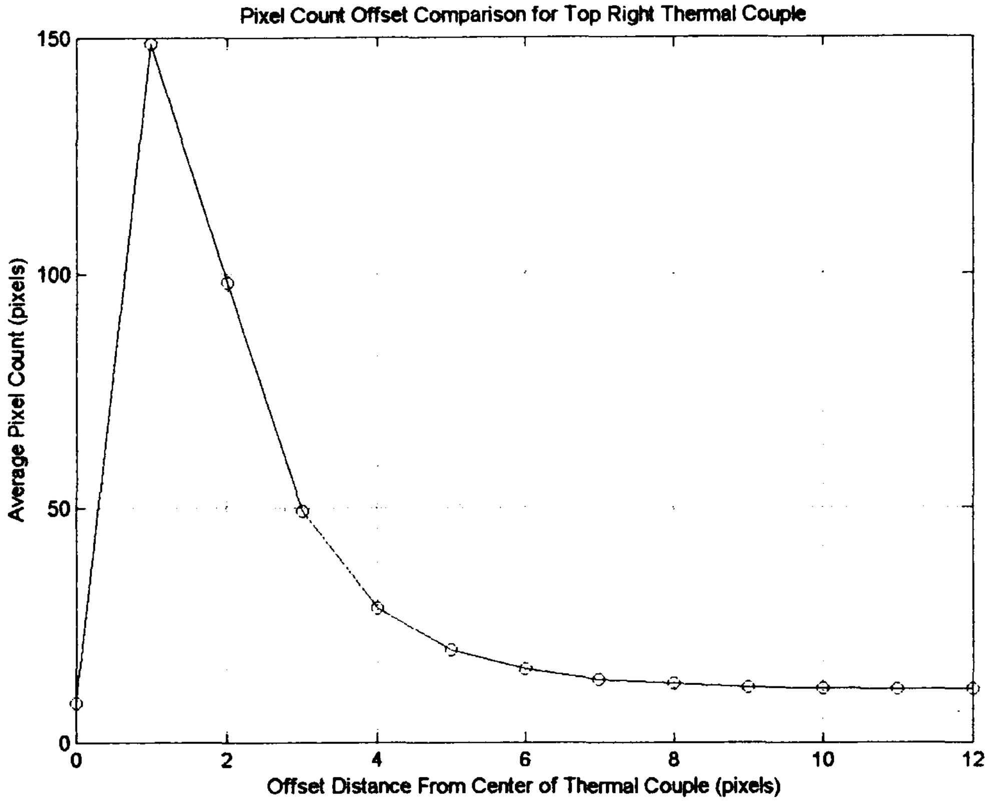

Fig. 9: Typical averaging results as a function of distance from the nominal top-right RTD #1 location.

As mentioned earlier, there is another element to consider to obtain IR data away from the painted surface. Namely, as the surface temperature rises during flight, the emittance characteristics of the paint change, thereby changing the affected radius of the surface coating. Examples of this are given on Figs. 10 and 11, which show zoomed IR image data (similar to Fig. 8) near the RTD #1 location at later times during the flight. Compared to Fig. 8, it is easily seen on these figures a variable radii size is needed to avoid the painted surface. To illustrate this change in radii numerically, a calculation similar to that used to produce Fig. 9 is used for the image data on Figs. 10 and 11. Figs. 12 and 13 show the result of the calculations, namely, the average pixel count as a function of radii away from the nominal center of the RTD #1 location. As seen, on Fig. 12, it takes about 4 to 5 pixel radius to get away from the painted surface, whereas in Fig. 13, it takes a radius of about 10 to 12. For this analysis, a variable radius would introduce a degree of complication that is not warranted. Thus, a constant value was used for all images.

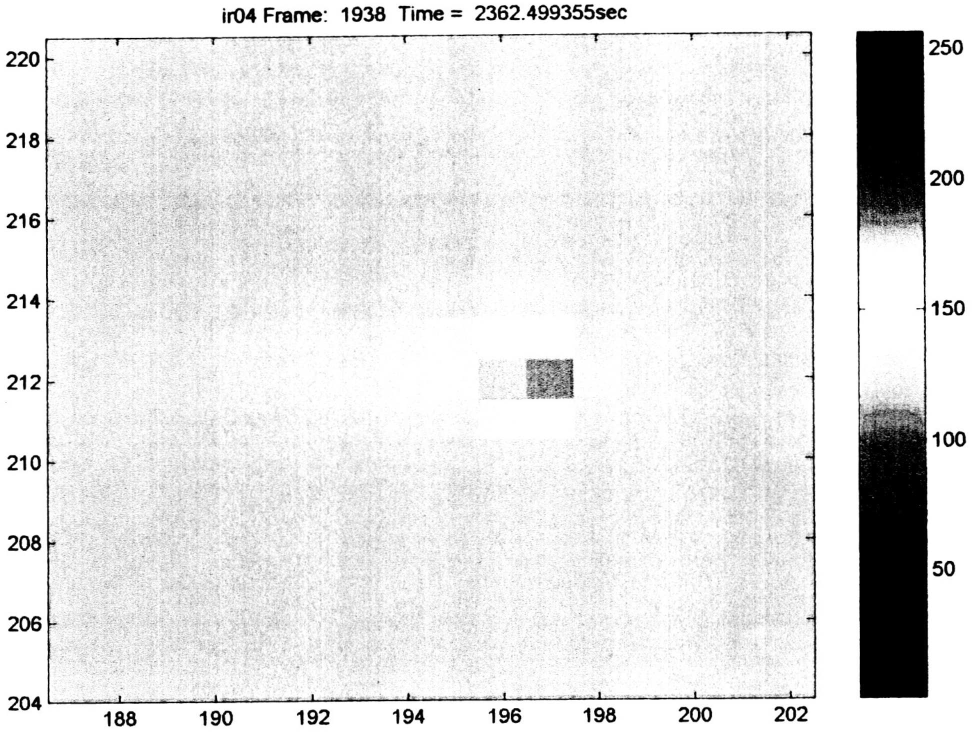

Fig. 10: Zoomed IR image data near RTD #1 at 2362.5 s

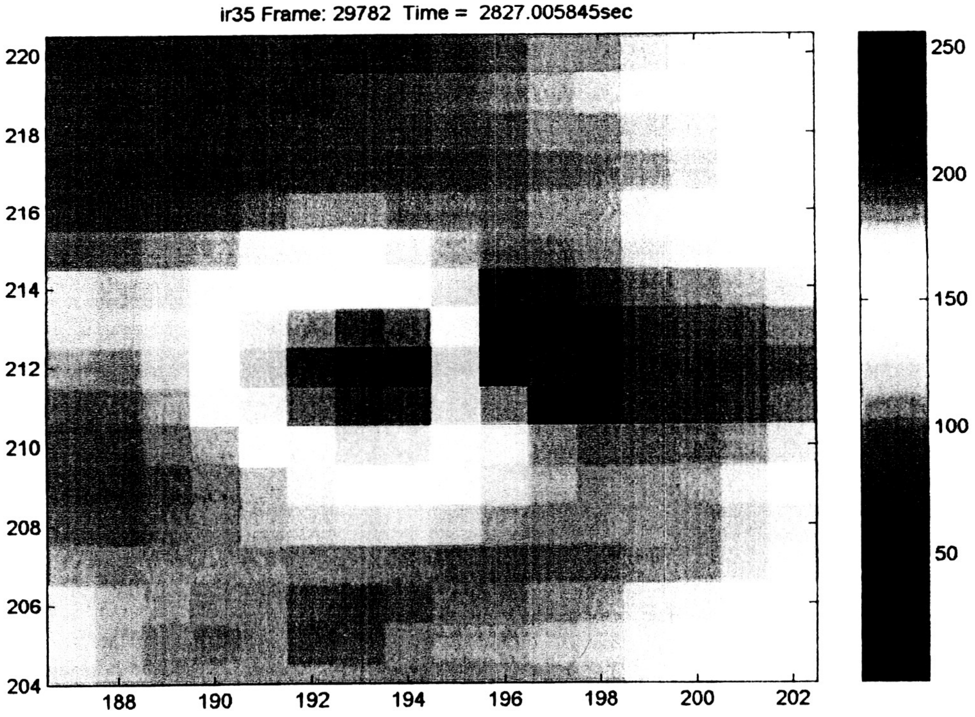

Fig. 11: Zoomed IR image data near RTD #1 at 2827.0 s

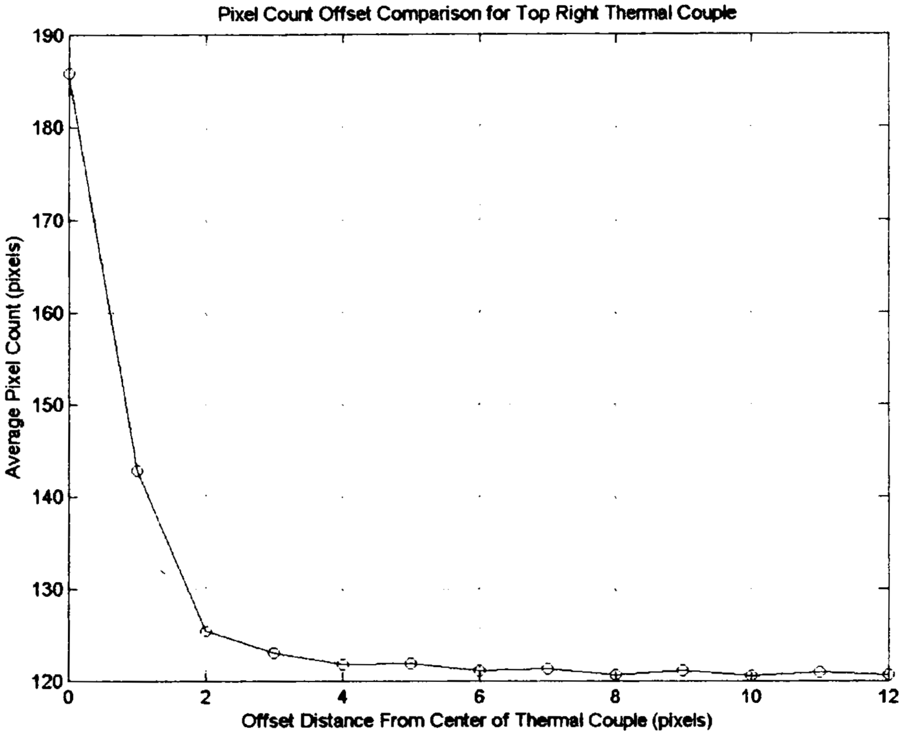

Fig. 12: Averaging results for Top-right RTD at t=2362.5 s

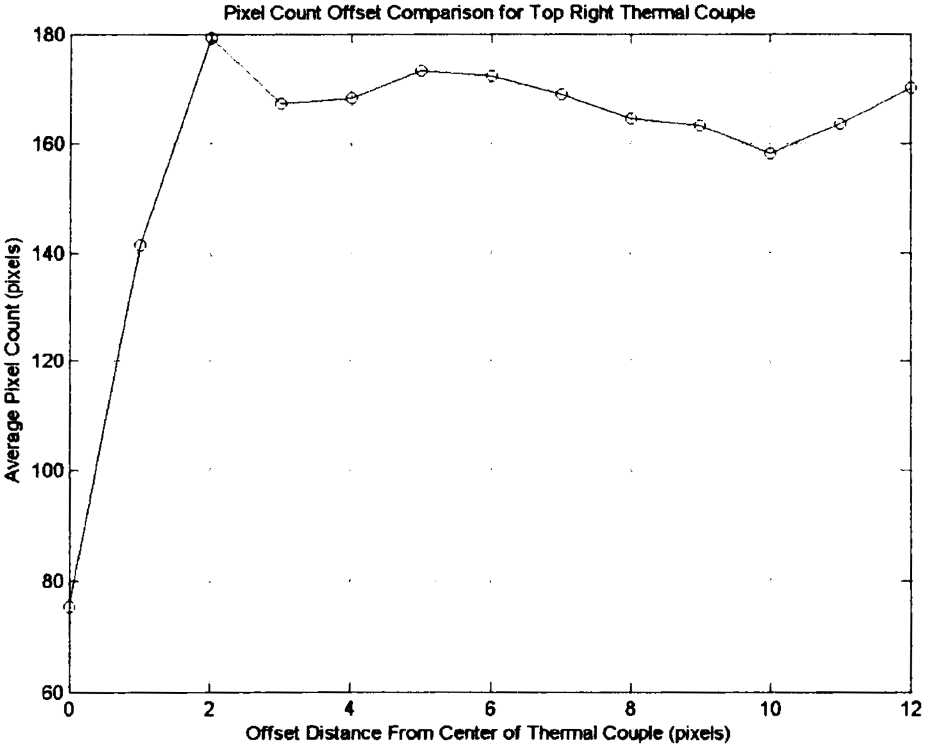

Fig. 13: Averaging results for Top-right RTD at t=2827.0 s

IR-RTD Comparisons

Fig. 14 shows a comparison of the IR image data at the four RTD locations to measured surface temperatures. The IR data have been adjusted to take auto scaling into account. As expected, there is a correlation between the IR data and the temperature as measured by the RTD sensors. Three of the four locations investigated show a distinct correlation, with the peak of the IR counts matching the maximum temperature from the RTD data. This correlation is nonlinear, and requires further investigation to establish the functional relationship. Also seen in the figure is an apparent problem with the IR image data at the top-left location (RTD #3). Comparison with the RTD data is not satisfactory. Initially it was thought that the IR system is "out-of-scale". However, the RTD temperature data does not verify this hypothesis (assuming appropriate calibration and a working sensor). Currently, the behavior of the RTD #3 sensor from 2400 s to 2600 s as shown in Fig. 4 appears unusual (rather low in value) when compared to the other sensors during this interval. This needs further investigation and coordination with DFRC personnel to resolve this issue.

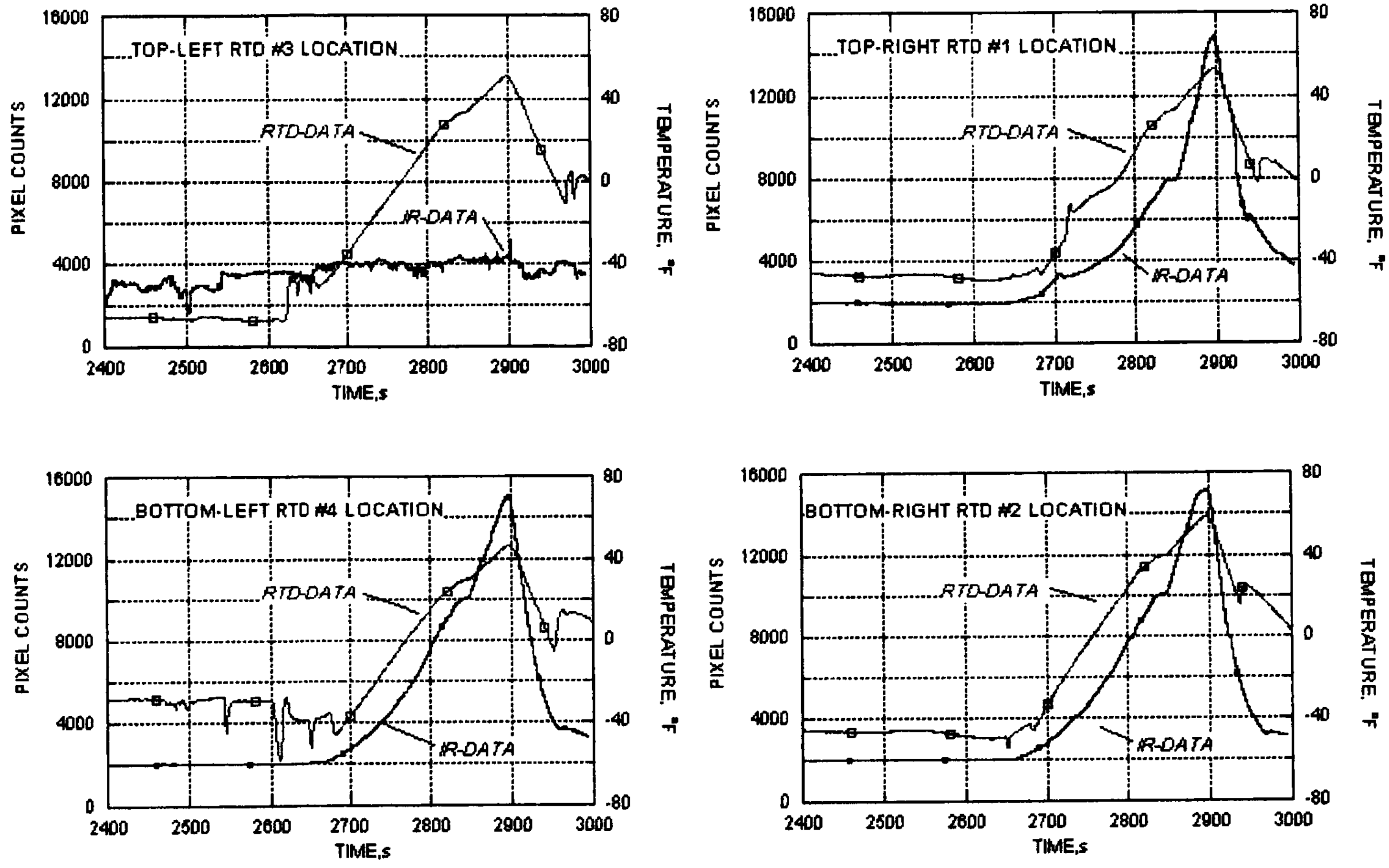

Fig. 14: Comparison of RTD and IR Data at all 4 RTD Locations

Conclusion

Analysis was done on IR data collected by DFRC on May 8, 2002. This includes the generation of a movie to initially examine the IR flight data. The production of the movie was challenged by the volume of data that needed to be processed, namely 40,500 images with each image (256 x 252) containing over 264 million points (pixel depth 4096). Initial examination provided useful information pertinent to the subsequent analysis. For instance, flight analysis can begin several frames from the initial time because there is no initial change in the data. Discussions with DFRC verified that the data on the first part of the files is taken in the hangar to check out the system and provide calibration data at room temperature. Movie "flickering" is a product of the movie generation process and will be dealt with later. However, discontinuities in the intensity color bar can now be explained by the camera auto ranging that took place during flight, and is not a flow phenomenon.

It was also observed during the initial analysis that the RTD surface coating has a different emissivity than the surroundings. This fact added unexpected complexity in obtaining a correlation between RTD data and IR data. A scheme was devised to generate IR data near the RTD location which is not affected by the surface coating This scheme is valid as long as the surface temperature as measured does not change to much over a few pixel distances from the RTD location.

After obtaining IR data near the RTD location, it is possible to make a direct comparison with the temperature as measured during flight (after the IR counts are "adjusted" to account for the camera rescaling effect). After adjustment, the IR data seems to correlate well to the flight temperature data at three of the four RTD locations. The maximum count intensity occurs close to the maximum temperature as measured during flight. At one location (RTD #3), there is poor correlation and this must be investigated before any further progress is possible. However, with successful comparisons at three locations, it seems there is great potential to be able to find a calibration curve for the data. Moreover, as such it will be possible to measure temperature directly from the IR data in the near future.

Future Work

To complete all of the research objectives, including obtaining temperature information from the IR measurements, require resolution of a few issues. One of the top priorities is to find out more about the camera operations, namely how it auto rescales in order to understand more thoroughly how to "adjust" the data. Additionally, the unusual IR data at the top-left RTD location must be resolved.

The issue here is, why the IR count data does not follow similar functional time behavior as the other three RTD locations. The cause may be possibly linked to large image angles between the camera and the surface normal. To determine this angle, the surface normal must be found using surface "grid" geometry coupled with the IR camera look angle. This lengthy, but necessary task must be undertaken to assure that observation geometry conditions are favorable. Once the issue is resolved, the next step is to correlate the four RTD temperatures with the adjusted IR counts to come up with "global" calibration curve. This may include exploring different functional relationships between the image count data and the temperatures as measured. Once this calibration process is developed, then there is the task of rescaling the entire image so that the calibration process can be applied to the entire surface of the fin to yield contour thermal maps Finally in addition in order to further validate the temperature results, a comparison can be made with CFD code results.

References

1. Throckmorton, D. A., Zoby, E. V., Dunavant, J. C. , and Myrick, D. L., "Shuttle Infrared Leeside Temperature Sensing (SILTS) Experiment-STS-35 and STS-40 Preliminary Results.", AIAA Paper 92-0126, January, 1992.

2. Davy, W. C., Green, M. J., "A Review of the Infrared Imagery of the Shuttle (IRIS) Experiment", NASA CP 3248 (part 1), Orbiter Experiments (OEX) Aerothermodynamic Symposium, April 27-30, 1993, Edited by D. A. Throckmorton

3. Blanchard, R. C., Wilmoth, R. G., Glass, C. E. , Merski, N. R. , Berry, S. A. ;Bozung, T. J. , Tietjen, A., Wendt, J., Dawson, D.: "Infrared Sensing Aeroheating Flight Experiment : STS-96 Flight Results". *Journal of Spacecraft and Rockets*, Vol. 38, No. 4, July-August, 2001, pg. 465-472.0

4. Banks, D. W., Van Dam, C. P., Shiu, H. J. and Miller, G. M. "Visualization of In-Flight Flow Phenomena Using Infrared Thermography," Proceedings of 9th International Symposium on Flow Visualization, Edinburgh, Scotland, Aug. 2000. (Also, NASA TM-2000-209027, July 2000)

Appendix

Acronyms

TR	Top Right RTD location
BR	Bottom Right RTD location
TL	Top Left RTD location
BL	Bottom Left RTD location
PC	pixel count
IR	Infrared
RTD	Resistive Temperature Device
DFRC	Dryden Flight Research Center
Fid	file identification number

Color Mapping

MATLAB color bars have a maximum of 256 different colors. The data acquired has a range of 0 - 4096 pixels. Thus the data requires a rescale to a range of 0 - 256 pixels. MATLAB's option to use direct mapping means specific pixel counts map directly to specific colors. Direct mapping is only performed on the data of the current frame being processed. MATLAB is constructed to directly map only the maximum and minimum pixel count data to 0 and 256 respectively. This means that each frame of the movie had its minimum and maximum data directly mapped to the color map colors represented by 0 and 256. However, note that MATLAB direct mapping is designed such that the data that falls in between the minimum and maximum values is linearly interpolated to intermediary color map colors. See Figure A-1.

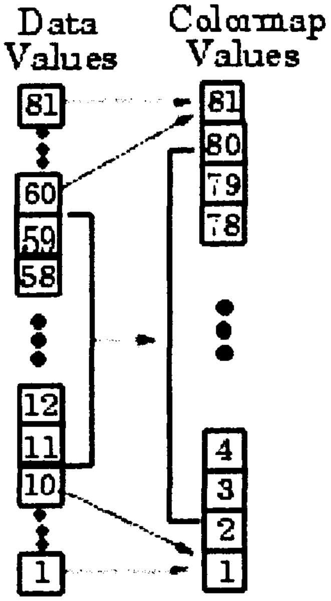

Figure A-1

Main Algorithm

MATLAB algorithm "IR2AVI.m" found at the end of the appendix is used to create the infrared movies and generate other outputs necessary in the analysis, such as IR pixel counts at specific locations. The algorithm works by opening one file at a time performing all the operations to that file that are required (i.e. to prepare the image data for the movie and generate pixel counts at all of the RTD locations).

IR2AVI.m creates an ".AVI" movie from the "ir n" (where n = 01,02,etc.) data files received from DFRC. The algorithm commands collect all necessary pixel count, time, and frame data at the specific RTD locations. These locations correspond to the drawings of the fin provided by DFRC

A description of the algorithm, and MATLAB commands, and steps used in the algorithm is found below.

The variable **"start"** is the variable that stores the file number. This is used to determine which file to start collecting data from. The algorithm then indexes through the remaining files while collecting data.

The variable "**file**" is used to recreate the file names of files that need to be opened.

There is a header that is 4kbytes in size between each image data set. After the header's time and frame data has been read into variables, the algorithm repositions its file identifier at the beginning of the file and counts in the 4096 data points to the beginning of the image matrix. The first part of each file contains the header information, which is read off and saved as a variable so that it can be displayed in the title of the movie later on.

The variable "**fid**" is a file identifier, which gets assigned to the data upon opening the file.

The function "**fseek**" allows the user to fast forward or rewind a specified number of data points.

The function "**fread**" allows the user to read off a specified number of data points into a variable.

There are 900 images in a file, and each image consists of a matrix of pixel counts that is 252 by 256 data bits in size. MATLAB reads the image matrix data as unassigned 16 bit integers and saves this data in a matrix ("**im**"). One image at a time is read off the file and displayed on MATLAB's figure window to be captured as a frame in the movie. Each file can be made into one 15 second long movie at 60 frames per second.

PIXCNT Function

The raw image data has pixel counts that vary from 2 to 4094. The raw image data matrix is loaded into the algorithm "**pixcnt**" which is also located at the end of the appendix. Pixcnt looks at "circles of data" finding the average pixel count of each circle. These "circles" represent the IR data about the RTD location. This analysis is necessary to avoid the complication associated with the paint emissivity at the RTD location. To illustrate this Figure A-2 is the image data matrix showing the pixel counts around the top right RTD at frame 84.

10.625	10.125	11.375	10.375	11.375	10.625	11.625	11.375	12.125	11.375	11.875	11.625	11.875	10.375	11.625	11.625	12.125	186
11.125	9.875	11.125	10.125	11.125	11.125	11.875	11.375	12.125	11.375	12.375	12.125	12.125	11.625	11.875	11.875	11.625	
10.875	10.375	11.375	11.125	11.625	11.125	12.625	11.875	12.875	11.875	12.625	12.125	12.625	11.875	12.375	12.125	12.125	
10.875	10.625	11.125	11.375	11.875	11.875	12.875	12.375	13.375	13.375	13.625	13.625	13.375	12.125	12.625	12.375	12.625	
10.875	10.875	11.125	11.375	11.875	12.375	13.375	14.375	15.625	15.375	15.375	14.375	13.625	13.375	12.875	13.125	12.625	
10.625	10.375	11.375	10.875	12.375	12.875	14.875	16.875	19.875	19.625	18.875	16.875	15.875	14.125	13.625	12.875	12.625	
10.875	11.875	11.125	11.125	12.875	13.875	17.375	22.625	28.375	28.875	25.625	20.375	18.125	15.625	14.625	13.125	12.625	
10.125	10.875	11.125	11.625	13.125	15.625	21.625	29.375	36.625	38.625	34.375	26.625	21.625	17.125	15.625	13.875	13.375	xdir |
10.125	10.875	11.625	12.375	14.125	17.625	26.125	36.625	47.875	53.375	48.875	37.625	28.125	21.625	17.125	14.875	13.625	
10.375	10.875	11.875	12.375	14.875	20.125	31.125	48.875	69.125	80.125	79.375	54.625	36.375	25.125	19.375	15.625	14.375	
11.125	10.375	11.875	12.625	16.875	23.625	40.875	82.375	154.63	215.38	155.88	80.875	45.125	28.625	21.125	16.125	14.375	
10.375	10.875	12.125	13.625	18.125	26.875	52.625	133.13	8.375	246.63	35.125	103.13	50.875	29.625	21.875	16.875	15.625	197
10.875	10.875	12.375	13.625	18.875	28.125	57.875	147.38	49.125	160.63	245.63	95.625	49.625	29.875	22.125	16.875	14.875	
11.125	10.375	12.375	13.625	18.375	27.125	49.625	99.125	193.13	203.63	126.38	68.375	41.375	26.625	20.875	16.875	15.125	
11.125	10.875	12.375	13.875	17.875	23.875	36.875	57.125	79.875	82.625	65.625	45.625	32.875	23.875	19.375	16.125	14.875	
11.375	10.875	12.125	13.375	16.125	19.875	26.875	35.125	43.125	43.375	39.125	31.125	25.625	20.625	17.875	15.875	14.375	
11.125	10.625	12.125	12.875	14.625	16.875	20.625	23.875	27.625	27.875	26.125	23.375	20.625	17.625	16.625	15.125	14.375	202
204						ydir ->pos		212								220	

Figure A-2: Data Matrix Around Top Right RTD at frame 84

This pixel count map corresponds to the image below; figure A-3 (note: figure A-3 is the same as figure 8 in the report, but rotated 90 degrees clockwise to match figure A-2 data). The axis labels on each figure are the same. Figure A-3 shows a zoomed in view of the top right RTD.

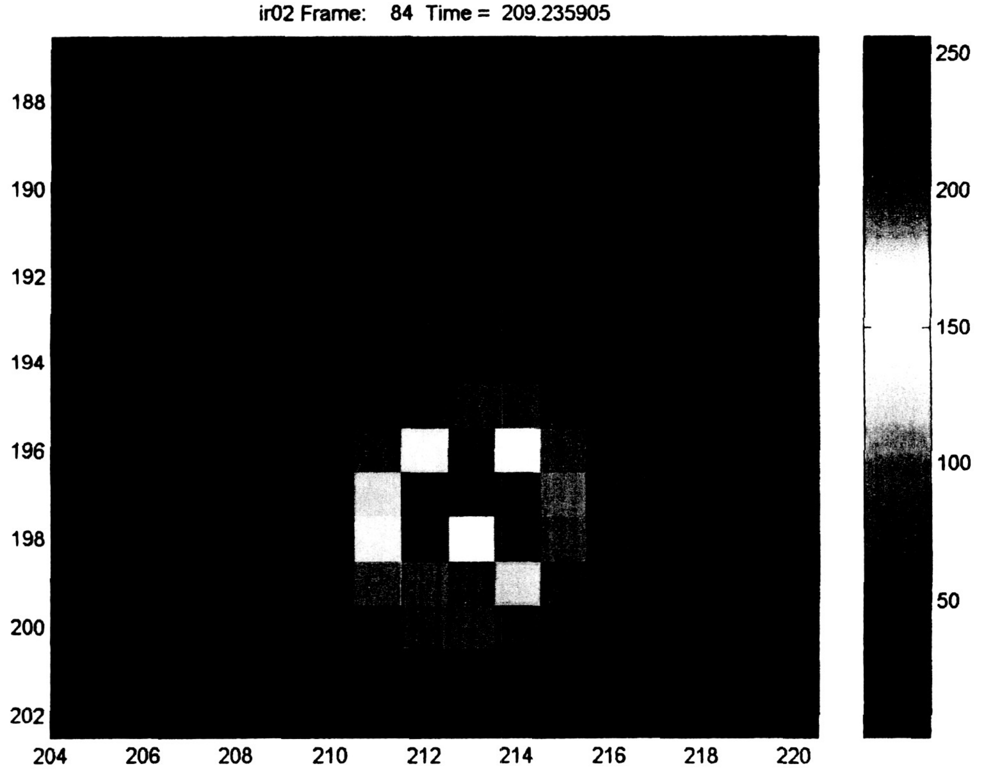

Figure A-3: Zoomed view of Top Right RTD at frame 84

Another simpler scheme was examined which involves taking four points at a given radius. That is a pixel count measurement is taken at a radius of four pixels, up, down, left, and to the right of the center pixel (i.e. a cross pattern). All four measurements are averaged and this is the average pixel count value that is used at the particular RTD location on all of the averaged pixel count plots in the report. This scheme has been compared with the "circle" averaging scheme previously mentioned and there is no significant difference. These average values are saved into the variable "**avgpcXX**", where XX stands for the appropriate RTD, i.e. TL, TR, etc.

Movie Generation

The image matrix,"**im**", is divided by 16 to give a maximum pixel count value of 256. It was found that the color map that was most illustrative was MATLAB's color map called "jet". The limits of the color map are manually set to 0 minimum and 256 maximum as discussed earlier. MATLAB's direct mapping is turned on and the color bar is added to the figure window.

The title, which includes the file number, the frame number and the time are displayed on the figure window. The axis of the figure window is set such that the y axis is positive in the upward direction.

MATLAB's function "**getframe**" is used to capture the frames to be put into the movie. Each frame is stored in a single column matrix "**I(i)**". The frames are then captured from a figure window, created previously, which is located at the dimensions "[LEFT BOTTOM WIDTH HEIGHT]" of the current figure window's axis system.

In order to cut down on storage in the active memory, files are now saved for the first time; they will be appended to later. The variable "**aviobj**" is created using getframe, this variable is to be used for combining the *.avi file with additional new frames. After this, the created files will be appended to this variable ("I(i)") on each time step (or iteration).

MATLAB's function "**movie2avi**" creates a *.avi by transforming the MATLAB movie matrix I(i) to ".avi" format at 60fps with the desired codec compressor and quality.

The memory is cleared and packed as much as possible, the open avi-object, and other open files are closed. The avi-object can now be reopened or loaded so that the next frame may be appended to it.

After all the frames from each file have been appended to the movie file and all the data has been collected similarly, the memory is totally cleared and the desired collected data is loaded for the plots to be created.

An additional function to create many of the plots used in the analysis is appended.

Data Manipulation

After the data is collected using the MATLAB algorithm, further adjustments are needed because of camera auto scaling. Microsoft's Excel is used to manually realign the data by selecting appropriate pixel values at several different times. This procedure can be automated provided the camera-operating algorithm properties are known.

The RTD temperature data used can be found in the file labeled flt0211-3RTD.exl.xls from DFRC. The time (tdig,sec), from this file is correlated to the time on the file "IRfull072602.avi".

Table A-1 lists the properties of the movie "IRfull072602.avi" which was used to come up with the time (tdig).

Table A-1: Properties of the Movie "Irfull072602.avi"

```
IRfull072602.avi
File Size: 929.69MB bytes
Total Duration: 0;11;00;20
Average Data Rate: 1.40MB per second
Image Size: 437 x 335
Pixel Depth: 24 bits
Frame Rate: 60.00 fps

AVI File details:
Contains 1 video track(s) and 0 audio track(s).

Video track 1:
Total duration is 0;11;00;20
Size is 928.78MB bytes (average frame = 24.01KB bytes)
There are 1416 keyframes, 38184 delta frames.
Frame rate is 60.00 fps
Frame size is 437 x 335
Depth is 24 bits.
Compressor: 'cvid', Cinepak Codec by Radius
```

IR2AVI.m

```
%% This M-file creates an ".AVI" movie from the "ir*" data files from DFRC.
%% The movie includes the background and time data
%% collects all pixelcount, time, and frame data at 4 thermal couple locations
%% these locations correspond to the blueprints of the fin from DFRC
%% By Matthew Carter 5/26/03 m.l.carter@larc.nasa.gov
clear all; close all; clc;
start=2; %% file number to start taking data from for creation of the movie
c=1; %% counter for index of frames and pixelcount
for n = start:45 %% Index through all 45 files
if n >= 1 <= 9;
file=['ir' num2str(0) num2str(n)]; %% used to recreate names of files
end
if n > 9;
file=['ir' num2str(n)]; %% recreates names of files
end
fid=fopen(file,'r'); %% Opens file and assigns a file identifier (fid)
fseek(fid,0,-1); %% rewinds (-1) fid to the very begining of file
f1=figure; %% so a figure isn't opened everytime figure is called
i=1; %% resets counter to 1(frame counter): 900 frames in a file
%%%% Creates movie frames from one file at a time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while i <= 900; %% 900 frames in a 15 second file (thats 60 fps)
%%%% Reproduces header text from binary file, to be displayed in title window
%% CHAR-> command used to convert the ASCI data into characters
%% STR2NUM-> converts string to number
F = char(fread(fid,5)'); %% reads first 5 bits of fid file into column F
fseek(fid,13,0);
T = char(fread(fid,12)');
fseek(fid,-30,0); %% resets to b4 time and frame data was taken
fseek(fid,4096,0); %% skips the 4k header in between each images' data set
im=fread(fid,[256,252],'uint16'); %% reads 1 frame into im
im=im'; %% unmodified transposed data (0 4095)
%%%% Gathers average pixel count data at specified locations
r=7; %% r-> radius of Thermocouple hole/region. in [pixels]
TR=(im(197,212+r)+im(197,212-r)+im(197+r,212)+im(197-r,212))/4;
BR=(im(184,119+r)+im(184,119-r)+im(184+r,119)+im(184-r,119))/4;
TL=(im(76,194+r)+im(76,194-r)+im(76+r,194)+im(76-r,194))/4;
BL=(im(125,117+r)+im(125,117-r)+im(125+r,117)+im(125-r,117))/4;
pixelcount(c,:)=[TR,BR,TL,BL];
%% Transform image data from [0 4095] to [0 256] scale for use with im... functions
%% There are 252 by 256 pixels each with an original depth of 256*16 pixelcounts
im=(im)./16; %% 4096/16=256 %% "." divide each element by 16
set(f1,'DoubleBuffer','on','Units','pixels');
colormap(jet(256)); %% sets current colormap to be jet colors
%% sets current axis Color limits properties cmin and cmax to 0 and 256
%% Data values in between 0 & 256 are linearly interpolated across the colormap
set(gca,'CLim',[0 256]); %% maps 0 data value to bottom of colormap
image(im,'CDataMapping','direct'), colorbar
title([file,' Frame: ', F, ' Time = ', T, 'sec '])
set(gca,'YDir','normal'); %% makes y-axis direction positive up from 0 ... 256
Time(c)=str2num(T); %% keeps track of displayed movie time
Frame(c)=str2num(F); %% records F frame number into Frame
F=F+1; %% counter for frames
```

```
%% GETFRAME -> used to capture the frames to be put on the movie
%% Each frame is stored in a single column matrix I(i)
%% The frames are captured from current figure window located at...
%% Dimensions.....[LEFT BOTTOM HEIGHT WIDTH] figure windows origin axis
if n == start;
%% with the transposed picture the WIDTH and HEIGHT parameters are reversed *********
I(i)=getframe(gca, [8 5 545 440]); %% includes the colorbar and title
if c == 1
save pixelcount.out pixelcount -ASCII
save Frame.out Frame -ASCII
save Time.out Time -ASCII
end
end
if n > start;
I=getframe(gca, [8 5 545 440]); %% gets a frame from current axis
aviobj = addframe(aviobj,I); %% combines avi files with more new frames
save pixelcount.out pixelcount -ASCII -append
save Frame.out Frame -ASCII -append
save Time.out Time -ASCII -append
end
c=c+1; %% total running counter for frame numbers
i=i+1; %% counter for frames in each file
%%%% Displays graph of average pixel count at increasingly larger diameters around TR TC
%% as a function of pixel distance from the center pixel of TR TC
if i == 8 & n == 2 | i == 62 & n == 4 | i == 6 & n == 35 %% data collect local
%%%% This is a check to make sure looking at the correct data
pause(3), axis([186.5 202.5 204 220.5]) %[204 220.5 186.5 202.5] %% zooms
%% grabs matrix of data surrounding TR TC (or the data in the current figure window)
IM=im(204:220,186:202); % (186:202,204:220); %
set(gca,'CLim',[0 256]);
pause(5), figure, image(IM,'CDataMapping','direct'), colorbar
%%%% Gathers data in order to display graph of PC ofset comparison around TR TC
[avgpcTR(n,:)]=pixcnt(im');
end
end
if n == start;
%% Creates an .avi by Transforming MATLAB movie to ".avi" format at 60fps
movie2avi(I, 'IR.avi', 'compression', 'Cinepak', 'fps', 60, 'quality', 100);
end
fclose('all'); %% closes all open file names
clear I; close all; pack;
if n == start;
%% opens old "avi" file so that it may have frames added to it
aviobj = avifile('IR.avi', 'compression', 'Cinepak', 'fps', 60, 'quality', 100);
end
fprintf('%g',n) %% file counter display for last file
```

```
end
clear all; %% Clears all variables from workspace
disp('done')
%%%% Loads data for plotting %%%%
load pixelcount.out pixelcount -ASCII
load Time.out Time -ASCII
plotterIR(pixelcount,Time,4095)
```

pixcnt.m

```
function [avgpcTR]=pixcnt(im)
%% This function finds the average pixel count in increasingly larger
%% cicles around the center pixel of the TR thermocouple
R=0:1:12; %% Range of offset from center pixel location of Thermocouple. pixels
%there are 252 by 256 pixels in im with a depth of 256 pixel counts
for i=1:13;
r=R(i); x=r; y=r;
%Gathers pixelcount data at specified locations
X=-x:1:x;Y=-y:1:y;
for a=1:length(X)
x=X(a);
for b=1:length(Y)
y=Y(b);
Value=im(197+x,212+y);
if b == 1 & a == 1
summ=Value;
else
summ=summ+Value;
end
end
end
Total(1,i)=summ; %% total circular region (everything inside)
if i >= 2
Summ(1,i)=Total(i)-Total(1,i-1); %% Just the circumference area (minus inside)
avgpcTR(1,i)=Summ(1,i)/(r*8); %%finds average
else
Summ(1,i)=summ;
avgpcTR(1,i)=Total(i); %% records average pixcnt to output variable
end
end
```

plotterIR.m

```
function plotterIR(pixelcount,Time,max)
%% max is the maximum number of pixelcounts of the data
%% pixelcount and Time must be the same size and there data should be from the same run
figure, plot(Time, pixelcount(:,1)),title('PC TR vs time (r=4)'),grid
xlabel('time (sec)');ylabel('average pixelcount');axis([2450 3000 0 max])
figure, plot(Time, pixelcount(:,2)),title('PC BR vs time (r=4)'),grid
xlabel('time (sec)');ylabel('average pixelcount');axis([2450 3000 0 max])
figure, plot(Time, pixelcount(:,3)),title('PC TL vs time (r=4)')
xlabel('time (sec)');ylabel('average pixelcount');grid
set (gcf,'PaperOrientation','landscape'),axis([2450 3000 0 max])
figure, plot(Time, pixelcount(:,4)),title('PC BL vs time (r=4)')
xlabel('time (sec)');ylabel('average pixelcount');grid
set (gcf,'PaperOrientation','landscape'),axis([2450 3000 0 max])
figure,
subplot(2,2,2), plot(Time, pixelcount(:,1)),title('TR'),axis([2450 3000 0 max]),grid
subplot(2,2,4), plot(Time, pixelcount(:,2)),title('BR'),axis([2450 3000 0 max]),grid
subplot(2,2,1), plot(Time, pixelcount(:,3)),title('TL'),axis([2450 3000 0 max]),grid
subplot(2,2,3), plot(Time, pixelcount(:,4)),title('BL'),axis([2450 3000 0 max])
xlabel('Time (sec)'); grid, ylabel('Pixel count');
figure, plot(Time,pixelcount(:,1),Time,pixelcount(:,2),Time,pixelcount(:,4))
title('TR BR BL Thermal Couple Locations at a radius of 4 pixels)')
legend('TR','BR','BL',2),axis([2450 3000 0 max]), grid
xlabel('Time (sec)');ylabel('Avg. Pixel Count');
for n=[2,4,35]
figure, plot(0:1:12, avgpcTR(n,:),'b-o');
title(['Pixel Count Offset Comparison for Top Right Thermal Couple Time = ', T, 'sec '] )
xlabel('Offset Distance From Center of Thermal Couple (Pixels)');
ylabel('Average Pixel Count (Pixel Count)'); grid,
end
```