NISTIR 5519

Visual Measurement Technique for Analysis of Nucleate Flow Boiling

Mark A. Kedzierski
Joseph M. Crowder
Anthony M. Jacobi
L. (Winston) Zhang

Building and Fire Research Laboratory
Gaithersburg, Maryland

NIST

U.S. Department of Commerce
Technology Administration
National Institute of Standards & Technology

Visual Measurement Technique for Analysis of Nucleate Flow Boiling

Mark A. Kedzierski
Joseph M. Crowder
Anthony M. Jacobi
L. (Winston) Zhang

October 1994

ABSTRACT

This report records a visual measurement technique for analyzing bubbles formed by nucleate flow boiling. The purpose of measuring these bubbles is to expand understanding the boiling process. By studying the behavior of these vapor bubbles, a better understanding of the physics controlling boiling heat transfer will be gained.

The boiling of refrigerants is simultaneously filmed with a high speed 16 mm camera while heat transfer measurements are taken. These tests are performed over a range of heat fluxes and flow velocities to study the effect of these conditions on the bubble dynamics.

Analysis of the 16 mm films evolved from a manual to a computer-assisted system, and finally to a digital image analysis system. The following is an overview of the filming apparatus and the bubble measurement schemes.

KEYWORDS

TABLE OF CONTENTS

NOMENCLATURE

English symbols

A	cross-sectional surface area of bubble (m)
a	major radius of bubble (m)
b	minor radius of bubble (m)
C_p	specific heat (J/kg·K)
D_i	Internal diameter of quartz tube (m)
D_s	equivalent spherical diameter (m)
fps	film speed (frames/s)
g	gravitational acceleration (m/s^2)
H	height of an image or object
H_i	height of film image projected by ideal system (m)
H_r	height of film image projected by real system (m)
K	vertical distortion factor (H_0/H_i)
h_{fg}	latent heat of vaporization (J/kg)
I	constants in optical correction eqn. (m^2)
N_e	number of end-bubbles for one trace
N_f	number of filming frames
N_z	number of zero-bubbles for one trace
P	vertical position
q_g	heat transfer due to bubble generation (W)
q_n	nucleate heat transfer (bubble generation & sensible) (W)
dQ_n/dN_f	heat transfer after bubble generation (W)
V_o	actual or object volume of bubble (m^3)
V_e	volume of all the end-bubbles for one film-trace (m^3)
V_z	volume of all the zero-bubbles for one film-trace (m^3)

Greek symbols

Δt	time difference between zero and end time lines (s)
Θ	angle between bubble major axis and r (deg.)
ρ	density (kg/m^3)
σ	surface-tension (N/m)

Subscripts

e	end
i	image, inner tube surface
o	object
adj	apparent image in tube

INTRODUCTION

This paper presents three methods used to visually study the boiling process. These visual studies lead to a better understanding of the fundamental mechanisms controlling boiling heat transfer.

Currently, there is no general mechanistic model to predict the heat transfer coefficient for boiling heat transfer. Empirical models exist for a limited range of fluids, heat fluxes, flow velocities, and geometries. Someday, a general mechanistic model will be used to accurately design and optimize refrigeration equipment. These visualization methods are used to add to the available mechanistic boiling data and to examine the applicability of existing correlations for predicting mechanistic information.

Test Apparatus

Figure 1 shows a schematic of the visualization test section. The test section consisted of a 9 mm internal diameter quartz tube with a 3 mm wall thickness. A thin (0.25 mm), 1.2 mm wide copper strip was located horizontally along the bottom of the tube with its length aligned with the flow direction. A direct current was passed through the copper strip to produce a range of uniform heat fluxes (q") from 15 to 30 kW/m^2. A plastic strip was cemented to the heater strip to insulate it and to be used in heat loss measurements. The localized heat flux produced nucleation only at the bottom of the tube. Bubble growth on the sides and top of the tube was not present which ensured an unobstructed view of the bubble growth on the tube bottom.
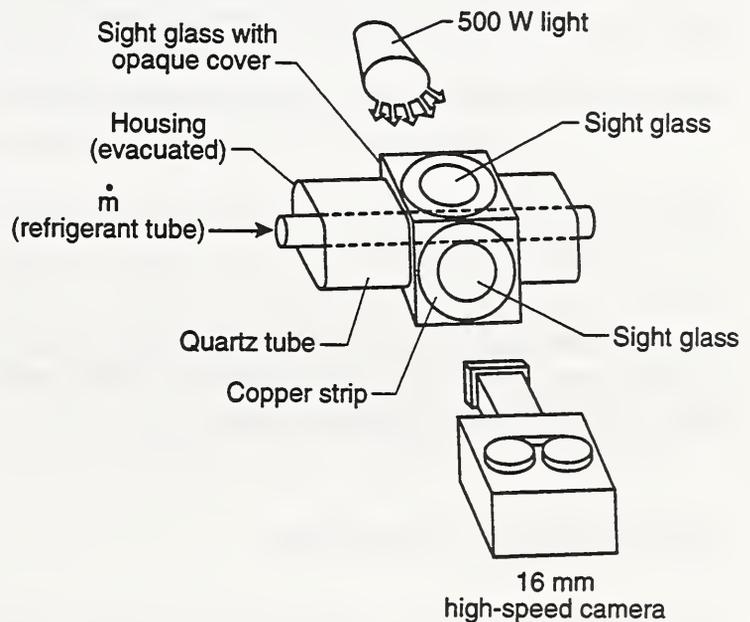
Figure 1 Schematic of visualization test section.

The quartz tube was enclosed in a safety housing with four flat windows located at opposite poles for lighting and filming of the boiling. The space between the housing and the quartz tube was evacuated to minimize the heat loss from the heater strip to the surroundings and promote one-dimensional conduction into the tube. The all-liquid Reynolds numbers based on the diameter of the quartz tube ranged from 0.0 to 9,500. The accuracy of the Reynolds number was 1.4-2.2% of the calculated value. The measurements for all fluids were obtained from the same smooth-tube test rig to ensure that observed differences were due to the characteristics of the fluids.

The inside surface of the quartz tube was roughened with a five micron polish to promote boiling. The roughness of the tube was measured, using a stylus instrument. The root mean square (rms) roughness (R_q), the Glättungstiefe (peak-to-mean) roughness (R_p), and the average

1

roughness (R_a) of the interior quartz surface were approximately 0.36 μm, 1 μm, and 0.3 μm, respectively. The R_a measurement is a good indication of the depth of the average cavity. Close examination of the roughness trace reveals inclusions of approximately 1 micron in width. Presumably, the average cavity is approximately 1 micron wide at the mouth and approximately 0.3 microns deep.

The heat flux incident to the quartz tube (q") and the temperature of the inner tube wall (T_{wi}) were determined from both measurement and theory. For example, the two-dimensional conduction within the plastic strip was modeled to determine the amount of energy transferred to the surroundings from the sides and bottom of the insulating strip. The model predicted a linear heat loss with respect to the temperature difference across the plastic. Consequently, the heat loss from the sides and bottom of the plastic was calculated from the measured temperature difference across the plastic. Next, the heat flux flowing into the quartz tube was calculated from the difference between the measured input power to the heater and the heat escaping from the sides of the plastic. The accuracy of the heat flux calculation was estimated to be approximately ± 1.7 kW/m^2. The estimated accuracy of the inner tube wall temperature (T_{wi}) calculation was $\pm$ 0.2 K. The T_{wi} was extrapolated from the measured temperature of the outer brass strip, using the calculated heat flux and a radial conduction model.

The two-phase heat transfer coefficient ($h_{2\phi}$) was calculated from:

$$h_{2\phi} = \frac{q''}{T_{wi} - T_s} \tag{1}$$

where the saturation temperature (T_s) was determined from the measured pressure to within 0.8 K. The accuracy of the heat transfer coefficient measurement for most of the data was estimated to be $\pm$ 10% of the measured value.

MEASUREMENT SCHEMES

The manual system for measuring the bubble volumes involved several steps. First, the film image was projected onto a large(1 m x 2 m) sheet of paper called a trace. The outlines of selected bubbles from several different frames were drawn on the trace. Next, the trace was taken to a drafting table where the area of each bubble on the trace was measured with a polar planimeter[1]. The major diameter and position of each bubble was measured with a caliper or a ruler. Finally, the measurements were scaled and corrected for optical distortion, and the heat load of each bubble was calculated. From the heat load of the bubbles, the total heat flux due to sensible and latent heat transfer was calculated.

This process was extremely time consuming and labor intensive. Time constraints minimized the number of frames from which data could be extracted. There was also a limit to the number of overlapping bubbles on one frame. If a few bubbles in one frame overlap, then the hidden parts of the bubbles could be approximated from parts of the edge that could be seen through

---

[1]   A polar planimeter is a mechanical device which measures the area of a plane figure on a piece of paper. Normally a pointer or stylus on the end of an arm is used to trace the edge of the figure, and an integrating wheel measures the area of the figure. [10]

the other bubbles, and by reversing the film and watching the bubble before it passed behind another bubble. However, this process became extremely difficult when large numbers of bubbles overlapped each other.

The computer-assisted system was developed as an intermediate method because the precision of the polar planimeter was limited by its clumsy mechanical mechanism. In this method, a hand-drawn trace was used, but the polar planimeter and ruler were replaced by a digitizing tablet and planimeter software.

The usefulness of the computer-assisted method was also limited by the maximum number of bubbles that could be measured on one frame (before overlapping becomes a problem) and by the slowness associated with such a labor intensive measurement. The advantages of this method were improved accuracy, improved reliability, and a significant difference in the amount of time required to measure all the bubbles on one trace (5-7 hours per trace as opposed to 8-10 hours). Due to efficiency, this is the method currently in operation.

Dr. Anthony Jacobi[2] received a grant to research the possibility of creating a digital image analysis system which would allow a totally automated bubble measurement scheme. The goal was to design a system whereby a single frame could be digitized using a CCD camera, and the software would pick out each bubble in the frame and give its measurements. This would facilitate examining more bubbles per frame and more frames from each film.

This project has met some of the goals, but has also experienced some problems. The system can identify and measure individual bubbles as long as the film has enough contrast and the bubbles do not overlap. When two bubbles overlap, a separate set of routines is used to extrapolate the size and shape of the two bubbles. Unfortunately, these routines are limited to only two overlapping bubbles with less than 40% overlap. When there are more than two bubbles overlapping or the two bubbles overlap too much, the routines do not work. This system also needs some human intervention in the filtering and thresholding procedures.

---

[2] At the start of the project, Dr. Jacobi was at Johns Hopkins University; he has since moved to the University of Illinois at Urbana-Champaign.

VISUAL MEASUREMENT OF BUBBLES

This section, an excerpt from [1], describes the steps involved in measuring the bubbles and calculating the volume, surface area, heat of formation, and heat load of the bubbles. Currently these steps are completed manually with the computer assistance. The Digital Image Analysis (DIA) method automates all of the steps.

DATA COLLECTION

Figure 2 depicts the bubble tracing method. The film-traces were used to calculate several quantities including the amount of energy transferred to the bubbles (q_n). Three different types of bubbles were defined to facilitate the calculation. The first set of bubbles, the zero-bubbles, are close to the heated surface under the zero time line. The time line is an arbitrarily chosen reference line below which all bubbles are analyzed. The time line moves so that the bubbles that were originally above the line at time-zero remain above it and newly generated bubbles are below it. The second set of bubbles, the end-bubbles, are all of the bubbles located under the time line after it has progressed from time-zero some finite time interval Δt. The difference between the volume of all of the end-bubbles and the volume of all of the zero-bubbles, along with the time interval Δt and the film speed, are used to calculate the q_n from:

$$q_n = \frac{(V_e - V_z)fps \; \rho_v h_{fg}}{N_f} \tag{2}$$

where fps is the film speed; N_f is the number of frames; ρ_v is the vapor density; and h_{fg} is the latent heat of vaporization; V_e and V_z are the total vapor volume at the end frame and zero frame, respectively. The third set of bubbles, the track-bubbles, are bubbles at discrete intervals of time along the time interval Δt. The track-bubbles are used to determine the amount of heat the bubbles receive or release after they have departed from the wall. This heat is used to correct q_n so that the heat used to generate the bubbles (q_g) can be calculated. The following equation was developed to approximate the corrected nucleative energy:

$$q_g = q_n - \frac{1}{2} \frac{dQ_n}{dN_f} fps \; (N_e - N_z) \tag{3}$$

where N_e and N_z are the number of bubbles in the end and zero frames, respectively; dQ_n/dN_f is the average rate of change in the energy received by the bubbles with respect to the number of filming frames. The magnitude of dQ_n/dN_f is equal to the slope obtained from a linear regression of a plot of q_n versus N_f for the track bubbles. The correction was positive nearly as many times as it was negative. The sole purpose of the track-bubbles was to obtain this rate of energy exchange for use in Eq. (3).

CORRECT FOR OPTICAL DISTORTION

The film images were corrected for optical distortions caused by the curvature of the quartz tube. Both the angle of the major axis of the bubble with respect to the axis of the tube (Θ) and its size were altered by the curvature of the tube, (see Figure 2). The aberrations could have been lessened by filling the cavity between the tube and the protective housing with a liquid having an index of refraction that closely matched that of the tube. However, this would have compromised the heat transfer measurements, since the amount of heat leaving the edges of the
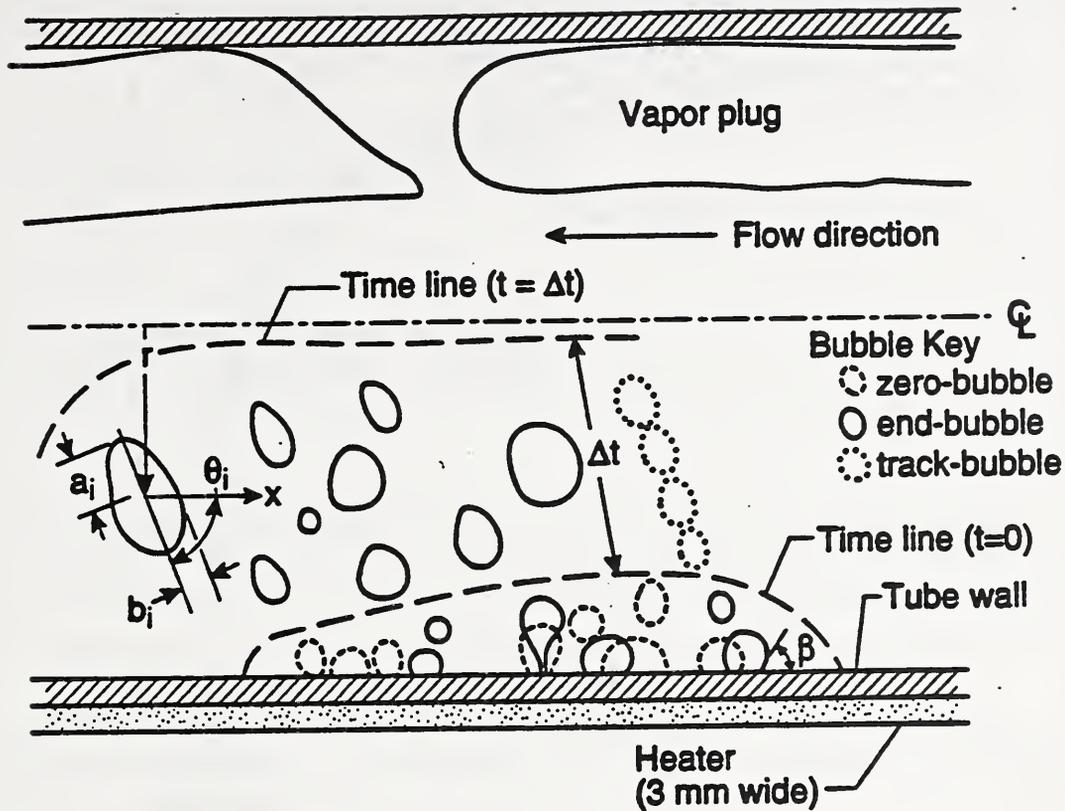
Figure 2 - Trace Method [1]

heater would have been increased by the presence of the liquid. Figure 3 shows the optical correction curves that were generated to correct the size of the bubbles after scaling the dimensions from the outer diameter of the tube. The ordinate is the ratio of the height of the image projected by the ideal system (H_i) to the height of the image projected by the real system (H_r). The abscissa is the vertical distance from the center of the tube. The real image is magnified by a relatively constant value with the exception of a small region very close to the tube wall where it is reduced. This distortion factor is calculated assuming that all of the bubble form on a line at the lowest point in the tube, and rise straight to the highest point in the tube. The real system is very close to this because the heater strip is very narrow.

It was necessary to determine the actual orientation of the bubble to calculate the actual length of the major radius (a_o) of the bubble. The following equation was developed [11] to find the true angle that the major axis of the bubble made with respect to the tube axis (Θ_o):

$$\theta_o = \frac{1}{2} arccot \frac{I_a - I_c}{I_b} \tag{4}$$

5
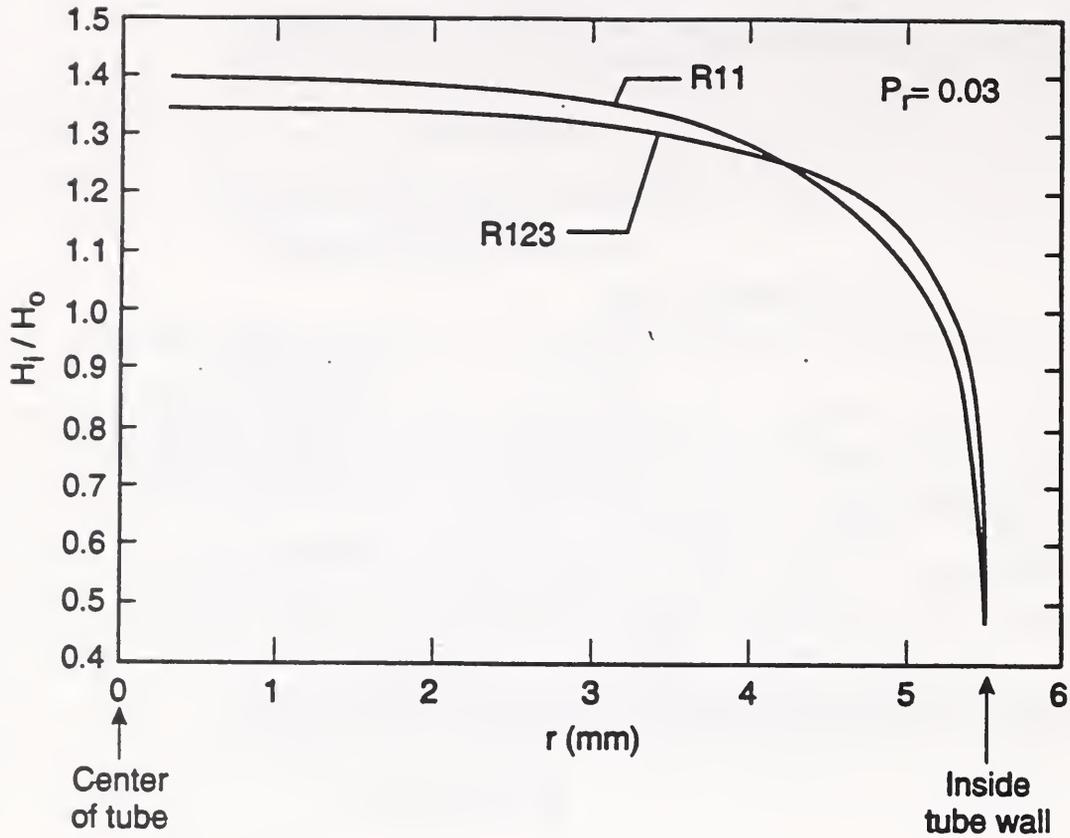
Figure 3 - Optical Correction Factor for R11 and R123 [1]

where

$$I_a = a_i^2\sin^2\theta_i + b_i^2\cos^2\theta_i, \tag{5}$$

$$I_b = 2K(b_i^2 - a_i^2)\sin\theta_i\cos\theta_i, \tag{6}$$

$$I_c = K^2(a_i^2\cos^2\theta_i + b_i^2\sin^2\theta_i) \tag{7}$$

a_i and the b_i are the minor and major radii of the distorted bubble. Θ_o and the Θ_i are the angles of the major axis with respect to the tube axis for the object and the image, respectively. K is the vertical distortion factor (H_i/H_r). Figure 2 illustrates the preceding bubble dimensions, where Θ_i is negative, since it is in the fourth quadrant. The bubble image is not rotated if $I_b = 0$, since the bubble cross section is circular.

6

The following equations relate the actual major and minor radii of the object (a_o and b_o) to the radii of the image and the actual bubble angle (a_i and b_i) [11]:

$$a_o = \frac{a_i b_i}{\sqrt{I_a \cos^2\theta_o + I_b \sin\theta_o \cos\theta_o + I_c \sin^2\theta_o}} \, , \qquad (8)$$

$$b_o = \frac{a_i b_i}{\sqrt{I_a \sin^2\theta_o - I_b \sin\theta_o \cos\theta_o + I_c \cos^2\theta_o}} \qquad (9)$$

One cautionary note: if the value obtained from Eq. (9) is larger than that obtained from Eq. (8) then Eq. (9) gives the value for the major radius and $\pi/2$ should be subtracted from the object angle which was obtained from Eq. (4). Otherwise, no corrections are necessary to the above optical equations.

The volume and the minor radius of the bubble were calculated from its major radius and the cross-sectional area of the actual bubble (A_o). A_o was obtained from the measured cross-sectional area of the bubble image (A_i) as:

$$A_o = A_i K^2 \qquad (10)$$

The major diameter was obtained by longitudinally splitting the area of the bubble in half with a line in the general direction of the major axis and measuring the distance between the two points of the line that intersect the bubble edges. It was assumed that surface tension and drag forces cause the bubble to be symmetric about its major axis. Accordingly, the bubble volume was obtained by revolving the cross section of the bubble about its major axis. The ellipsoidal volume was calculated from:

$$V_o = \frac{4A_o^2}{3\pi a_o} \qquad (11)$$

CALCULATION OF OPTICAL DISTORTION CORRECTION FACTOR

The film images taken by the high speed camera were distorted by several lenses between the film and the bubbles. Any interface between two mediums with different indices of refraction acts as a lens. This means that the inside and outside surfaces of the quartz tube were lenses. Unfortunately, the lens system could not be properly focused or compensated to produce an undistorted image on the film. Therefore, a correction factor for this distortion had to be applied to the measurements taken from the film. The process of finding that correction factor is described below.

BEAM 4 Version 4.56 is a light beam tracing program which is capable of plotting the path of a light ray emitted from any point at any angle through a set of lenses. The BEAM4 program was used to simulate the quartz tube and the camera to study the visual distortion effect of the quartz tube on bubble dimensions. The distortion was calculated by simulating evenly spaced ray sources inside the tube and finding where each of these ray sources focused on the film.

THE OPTICS

The real lens system included the refrigerant, the tube, the camera lens, the iris, and the film. The tube had an inner diameter of 9.017 mm an outer diameter of 11.455 mm and an index of refraction of 1.46. The idealized system was identical to the real system, but did not have the tube or the refrigerant. The idealized system does not suffer the distortion caused by the tube, so it was used as a basis of comparison to find the distortion of the real system. Figure 4 shows the simulated optics of the real system.
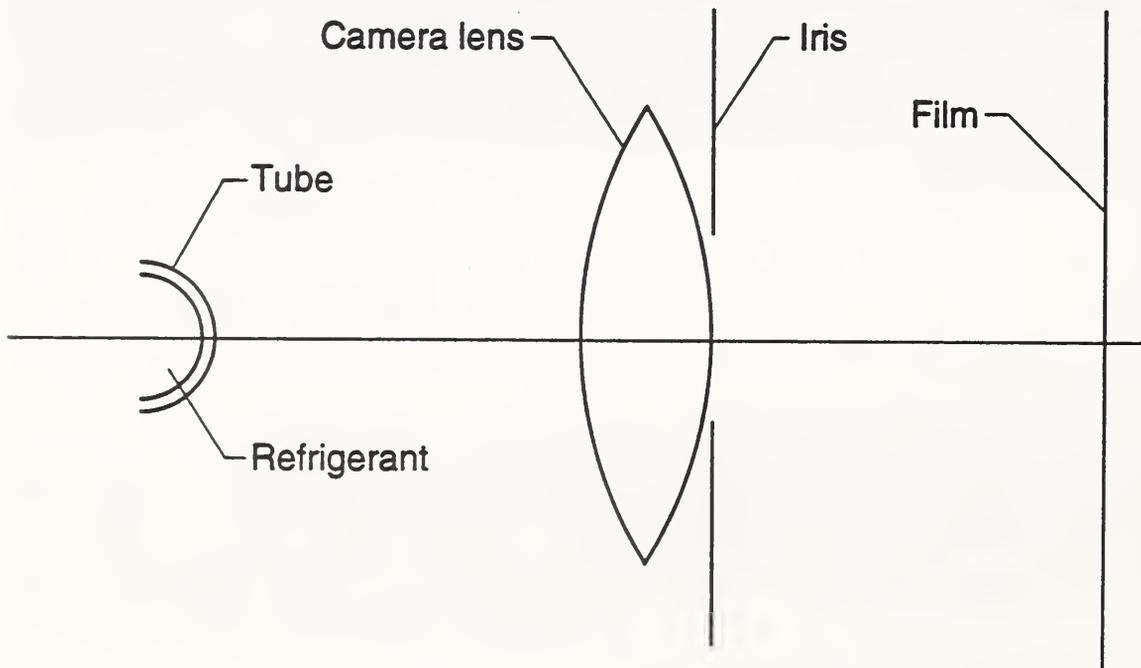
Figure 4 - The Optics

The camera lens was modeled as a simple lens and iris system. The actual lens used was much more complicated with a series of lenses designed to correct for the distortion caused by using spherical lenses. The simple lens model is sufficient because any distortion caused by this lens was cancelled when the results for the real system were compared to the results for the idealized system.

THE RAYS

The tube radius is divided into 17 segments with a test point at the end of each segment as shown in Figure 5. The reason for this was that the distortion factor is changing in an almost linear fashion closer to the center, but increases exponentially near the edge of the tube.
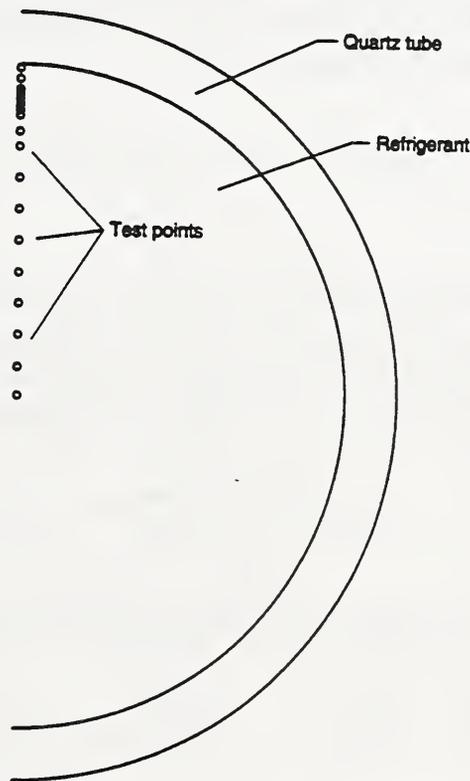
Figure 5 - Test points in the tube

<u>Focus</u>
The overall focal length was found by using BEAM4's autoadjust function to move the film horizontally as close to each test point's ideal focal point as possible. Finding the ideal focal point for a test point was a two step process. First, the single best ray path was found by trial and error. The best ray path was the one that passed through the exact center of the camera lens[3]. Then six more rays were added starting at the test point and offset by 0.005 radian from

---

[3]   The best ray path passes through the center of the lens because that is the point of least spherical distortion. Using rays that passed through the edges of the lens as well, as the center, would make finding the focal point impossible, because not all of the rays intersect the film at the same point. Most cameras use an iris that is small compared to the size of the lens to let in only the light rays that focus sharply on the film.

9

each other. The focal point was the point where all of the rays originating from one point and passing through a lens system intersect. The film was located at the focal point to get the sharpest image. This is not the way a real camera is usually focussed; the lenses are moved instead of the film to focus. The distance travelled by the lens or film is extremely small so this difference does not cause any problems.

CORRECTION FACTOR AND ADJUSTED POSITION

The location of the film image for each point inside the tube was found by averaging the point of intersection with the film for the seven rays from each test point. With this data for the two optical systems, a correction factor and an adjusted position was calculated for each test point.

The correction factor (H_i/H_r) is a ratio of the size difference between the images of the same object projected by the two optical systems. The distance between two test points was used as the object size, so vertical distance between the images of those two test points was used as the image height. H_i/H_r was calculated for each segment:

$$\left(\frac{H_i}{H_r}\right)_{ab} = \frac{P_{i_a} - P_{i_b}}{P_{r_a} - P_{r_b}} \qquad a \text{ and } b \text{ are the end points of a segment } ab \qquad (12)$$

The adjusted position was calculated so that a correlation can be made between the correction factor and the apparent position of the object in the tube. The apparent position of the bubble in the tube can be directly calculated from the measurements on the trace. Calculating the actual position of a bubble in the tube would require an iterative approach.

$$P_{adj_{ab}} = \frac{P_{r_a} + P_{r_b}}{2} \times \frac{P_{adj_{max}}}{P_{r_{max}}} \qquad (13)$$

P_{adj} was the apparent distance from the center of the tube to the center of the line segment. This distance was scaled, but not corrected for optical distortion.

A useful correlation between the correction factor and the position was:

$$\frac{H_i}{H_r} = F(P_{adj}) \qquad (14)$$

Since the correction factor was not linearly related to the position, a function fitting program was used to fit an appropriate function to the values of the correction factor. This resulted in an exponential form for the points near the edge and a quadratic form for the points nearer to the center.

$$\frac{H_i}{H_r} = c_1 x^2 + c_2 x + c_3 \qquad x < r$$

$$\frac{H_i}{H_r} = c_4 e^{c_5 x} + c_6 x + c_7 \qquad x \geq r \qquad (15)$$

Where x is the distance from the center and r is the critical radius for the curve fit.

The manual system section describes the process of tracing bubbles from high-speed film. A paper trace of the bubbles is necessary to find the surface area and volume of the bubbles. The paper trace contains red zero bubbles, black end bubbles, and green track bubbles.

The difference in volume between the zero and end bubbles provides the amount of vapor formed within the time interval between the Start Frame and the End Frames. The track bubbles are used to measure the amount of volume change after the bubble has been released from the wall. This permits the calculation of the amount of sensible heat transferred to the bubbles. The sensible heat is subtracted from the entire energy found from the zero and end bubbles to give the amount of energy required to generate the bubbles at the wall, so that the amount of sensible and latent heat transfer can be calculated separately. The top and bottom of the tube are drawn on the paper trace to scale the drawn bubbles to their true size. Figure 6 shows an example of a paper trace.

Figure 6   A Paper Trace

EQUIPMENT

A 16 mm film analyzer is used for stepping through the films. A film analyzer is a special type of projector with the capability of stepping through a film at different speeds or stopping at a single frame. The analyzer is set up in a darkened room with a paper trace taped to the wall as a screen. The entire film image fits within the 1 x 2 meter paper trace.

SETUP

First, the 16 mm analyzer was loaded with a film and the paper trace was taped to the screen. Next, the film was advanced a few hundred frames. The top and bottom of the tube, and the bottom of the heater strip were drawn on the trace. These three lines were drawn from a frame near the beginning because the outer tube walls were easier to see near the beginning of the film. These lines were used to scale the measurements (the size of the projected image was not always the same for each film). The refrigerant, film number, and date were annotated in the upper left corner.

The film was advanced to a point where the red timing dot on the right of the film was on every sixth frame and not moving too much[4]. If the dot was moving, more measurements were made after all the bubbles were traced, to find the exact film speed.

TRACING BUBBLES

Red Bubbles

  Choosing A Zero Frame. The zero frame is the one from which the zero bubbles were traced and the time reference t=0 was established. A frame was selected where: (1) the timing dot indicates that the camera was close to 6000 frame per second, and (2) the bubbles near the bottom of the tube were discrete and isolated. The nearest frame with a four digit number on the left side of the frame was chosen to be the zero frame. The top of the paper was marked "RED @ (frame number),"[5] and the frame counter of the analyzer was reset.

  Choosing The Bubbles. The right and left limits of the trace along the bottom tube wall were selected. The limits were selected based on which sites were discrete and visible by cutting off any sites that were not totally in the frame or that produced a lot of bubbles that converged with bubbles that were formed outside of the limits. Also, any sites which produced more than a few bubbles which leave the frame before the End Frame were eliminated.

A line was drawn through the convenient break that was chosen. This was the zero time line which was mentioned in the section on visual measurement of bubbles. The bubbles below this zero time line were the red bubbles. The bubbles above this line were ignored. The film was advanced and observed to see if any red bubbles merged with bubbles that started outside of the limits or above the limit line within 30-50 frames. If so, then the limits or the zero time line were adjusted to include both bubbles or exclude both bubbles, whichever was more convenient.

  Drawing The Bubbles. The edges of all the bubbles below the zero time line were traced with a red pencil. While drawing them, it was necessary to roll the film back and forth a few times to watch their shapes develop, but the bubbles were only drawn from the Start Frame. If two bubbles overlapped, a dotted line was drawn for the edges that were hidden. Sometimes,

---

[4]   The timing light flashes once per millisecond, so when the dot is on every sixth frame the film is moving at about 6000 frames per second.

[5]   The frame number is the number on the side of the film, not the number from the frame counter. Sometimes this number is obscured by the analyzer. In this case, the film was moved a little in the analyzer in order to read the number.

part of an edge could be seen through another bubble. That edge was used as a guideline if the apparent edge matched expectations. Some of what appeared to be edges were only reflections of other bubbles or the side of the tube so caution and intuition were used to draw the edges. The red bubbles are shown with dashed lines in Figure 6.

Black Bubbles

Choosing The End Frame. The film was advanced past the Start Frame until some of the bubbles which started below the zero time line were close to going off the film or touching the top of the liquid-vapor interface. The red bubbles and all those bubbles formed within the limits subsequently were enclosed below the advancing time line. If a few bubbles went off the film relatively quickly but the rest stayed, then the film was advanced about 150-200 frames past the Start Frame or until just before a lot of bubbles get into the unusable range. This frame was selected as the End Frame. Mark at the top of the trace how many frames past the Start Frame this is.

Drawing The Bubbles. Using the same methods as before, the black bubbles were drawn on the same trace. Only the bubbles that were red bubbles or that formed since the Start Frame inside of the limits that were set up were drawn. Any bubbles that were above the zero time line at the Start Frame or that were formed outside the limits were ignored. The film was rewound to the Start Frame several times to identify which bubbles to draw.

Some bubbles went off the film or merged with unusable bubbles before the End Frame. When this happened, a blue pencil was used to trace them just before they become unusable. Each blue bubble was marked with the frame number. No more than 10 or 12 blue bubbles appeared along the top and left sides. Because the flow was right to left, some bubbles flowed off the left side. Several blue bubbles were scattered around that were marked just before they merged with a bubble which was above the time line. If two usable bubbles merged, no problem existed. The final bubble was drawn in black at the End Frame. The black and blue bubbles are shown with solid lines in Figure 6.

Green Bubbles

The green bubbles were representative bubbles that were traced every 10 to 20 frames. They were used to study the action of the bubbles after formation and to determine whether the liquid was superheated or subcooled.

A bubble that does not interact with other bubbles between the Start Frame and End Frame was chosen. Starting at least a few frames after the bubble was fully formed, the bubble was traced in green. The letter code and the number of frames past the zero frame (ex. A@20) was marked inside the bubble. The film was advanced 10 to 20 frames; the same bubble was traced again and marked with the same letter code and the new frame number. This was repeated until the bubble had been traced at least six times. The whole procedure was repeated for at least four separate bubbles. If a bubble that was traced as a black or red bubble was used, that black or red bubble was also marked with its letter code and frame number. The green bubbles were used to study the growth of independent bubbles after formation, so care was used to ensure that none of the green bubbles interacted significantly with any other bubbles, and were not drawn during bubble formation. The green bubbles are shown with long dashed lines in Figure 6.

13

ANNOTATING THE TRACE

Number of Sites
After all of the red, black, and green bubbles were drawn, the boiling sites were found and marked. The film was rewound to the Start Frame then slowly advanced and the approximate number of sites in each area of the tube bottom was marked on the trace. The sites tended to be clumped together to such a degree that it was difficult to identify and count individual boiling sites. When this occurred, a range of numbers was given (e.g.: 1-2, 5-7).

Film Speed
If the timing dot was not stationary between the Start and End Frames, then the exact film speed needed to be calculated. Once all of the drawing was completed, a numbered frame between the Start and End frames was found. That section of film was pulled out of the analyzer and the distance between two dots on the film and the length of seven frames was measured using a good pair of dial calipers. These two measurements were marked on the trace and the film speed was calculated using the following formula:

$$fps = \frac{1000 \; dots}{1 \; s} \times \frac{7 \; frames}{x \; mm} \times \frac{y \; mm}{1 \; dot} \tag{16}$$

Bubble Numbers
The bubbles were numbered for tracking and checking the measurements. Each color of bubbles was separately numbered to reflect a black #1, a red #1, and a green #1. The order that the bubbles were numbered in is not important, but a logical progression tended to save time in the following steps.

Major Diameters
A line was drawn on the major axis of each bubble. This major axis was along the longest axis, and symmetrically split the area of the bubble into two equal halves. If the bubble was not close to elliptical in shape, this was difficult to do. In this case, the major axis was drawn so that it split the area evenly and it was as close to the longest diameter as practical.

Measuring The Bubbles
The length of the major diameter, the distance from the bottom of the tube, and the area of each bubble was measured and recorded. The area was measured with a polar planimeter, and the major diameter and distance from the bottom were measured with a ruler.

Data Reduction
A data file was made for the data reduction program VISDAT to read. The version of the data reduction program VISDAT which was used for this method is no longer in existence. The current version is adapted for use with the computer-assisted system. However, the procedure for using it was similar to that for the computer-assisted system.

COMPUTER-ASSISTED SYSTEM

The computer-assisted system is identical to the manual system except that the planimeter is replaced by a digitizing tablet and a computer with planimeter software. Measurements with the computer-assisted system are significantly quicker and more accurate.

Drawing and annotating the trace for this system is the same procedure as for the manual system. The only difference is that a line needs to be drawn exactly ten inches above the bottom of tube line. This line is used to locate the map for the upper bubbles. See the section on making diameter files for an explanation.

EQUIPMENT

A CalComp digitizing tablet is used as an input device for an IBM compatible computer running EasyDIJ. A thin (1/8") plexiglas sheet approximately the same size as the digitizing tablet is placed on top of the trace when it is on the digitizer to keep the trace from bunching up or sliding while the cursor is used to trace the bubbles.

EasyDIJ is a planimeter program. This program takes position input from the digitizing tablet, and uses it to draw a figure on the screen, and calculate the area and circumference of the figure.

SETUP

Directory and File Setup
Once the computer is on and the digitizer tablet connected and turned on, move to the C:\EASYDIJ\DATA\ directory. Then enter the trace directory for the name of the working trace (i.e. R123.1, R11.27). The commands are:

```
C:
CD \EASYDIJ\DATA
CD [trace directory name]
```

If a directory does not already exist, make a new directory and copy a few files into it. Enter the following commands from the C:\EASYDIJ\DATA directory:

```
MD R[refrigerant no.].[trace no.]
CD R[refrigerant no.].[trace no.]
COPY C:\EASYDIJ\DIG.AAA
COPY C:\EASYDIJ\DATA\R[refrigerant no.].VAR
RENAME *.VAR R[refrigerant no.]#[trace no.].VAR
```

Type the following commands to enter EASYDIJ:

```
EJ
7014  (if black screen comes up)
```

The license number (7014) is required when a new trace is made or an old trace is modified on a new day.

These instructions assume that the user is working on the computer which has established data and easydij directories. If not, then the data directory will have to be made, and the easydij directory put into the path statement in the AUTOEXEC.BAT file. The VAR file will also have to be copied into the data directory.

Locating The Map

Locating the map is required to ensure that the position, angle, and length of the major diameters are correct. To do this, enter the Locate Map menu and set the coordinates for three points (A, B, and C), then tell the computer where those points are on the digitizer. By defining three points on the digitizer board, the rest of the points on the board are defined. The commands are as follows:

L   (locate map menu)

N   (do not change the coordinate system)

A 0 0   (set the A coordinate to North=0, East=0)

B 8 0   (North=8, East=0)

C 0 8   (North=0, East=8)

X   (digitize point A)
        Put the curser cross hairs on A and click button #0

Y   (digitize point B)
        Put the curser cross hairs on B and click button #0

Z   (digitize point C)
        Put the curser cross hairs on C and click button #0

Q   (quit from Locate Map Menu)

MEASURING

Making Area Files (3 Per Trace)

Use the planimeter function to make a file containing all of the bubble areas. Make one area file for each color of bubble on each trace (B.ARE, G.ARE, and R.ARE).

It is very important that the bubbles are digitized in order. If they are out of order, the major diameters will correspond to the wrong area. This can severely skew the results.

It is also important to digitize slowly and smoothly. EASYDIJ accepts data relatively slowly and uses a straight line approximation between digitized points. If the cursor is moved too quickly the distance between points grows and the inaccuracy of the approximation increases. Additionally, a slow and steady movement of the cursor will prevent a slip or stutter from causing a large error.

To start taking area measurements, first ensure that the map has been located. Then position the trace on the tablet with the first bubble inside the active area of the tablet. The exact position does not matter at this point because only the area is being measured, not the position. Place the 1/8" plexiglas plate over the tablet and trace to keep the trace from sliding or bunching up while digitizing. Execute the following commands:

D   (enter the digitize menu)
P   (planimeter)

enter the filename [first letter of color].ARE (ex. B.ARE) and press enter twice.

W   (write over old file if it exists)
Y   (plot the points on the screen)

At this point the planimeter menu appears. Make sure that the scale is set to 1"=1".

+   (add an area)

Starting with a mark on the circumference, trace black bubble #1 with the curser cross hairs. Each tick from the computer is a point on the circumference. When the circumference has been completed, stop just short of the starting point. If the same point is digitized over again, the area recorded may be larger than the actual area.

+   (planimeter menu will flash for an instant then return to a blank graphics screen)

Trace black bubble #2 and press the [+] key. Trace each black bubble in order until the last bubble is finished. After the last bubble press <esc> instead of [+]. This will return the planimeter menu which gives the option to quit to the main menu. Repeat the whole procedure for each color of bubble.

If the cursor slips while tracing a bubble, or the wrong bubble is traced, press the [-] key instead of the [+] key, and start over on the correct bubble (or repeat the same bubble if the cursor slipped). Later, when the data is double-checked, the flawed record can be removed.

Making Diameter Files (1 per bubble)
Use the list function to record the end points of the major diameter. Make one file for each bubble (ex. B.003, G.027).

Since a diameter file is made for each bubble, the order in which they are made is not irrelevant. However, the position of the trace on the digitizing tablet is very important because MAJDIA uses the North position to determine the distance from the bottom of the tube. The position is important because the optical distortion changes as a function of distance from the bottom of the tube (see the section on optical distortion correction factor).

Position the trace on the tablet so that the bottom of the tube line is exactly superimposed on the line from A to C (the horizontal position doesn't matter, only the vertical position and angle). Get into the List function as follows:

D   (digitize menu)
L   (list)

17

N or G   (display in numeric or graphic form. It does not matter which is used)

Enter the filename as follows:

[first letter of color].[bubble number]

The bubble number must be a three digit number or MAJDIA will not recognize it e.g., R.001 works, R.1 does not.

N   (no constant elevation)
W   (write over old file)

One tick should sound. If the software is in numeric mode, it will say Ready For Point 1. Digitize the two end points of the major diameter. At this point pressing <enter> will put the system back into the list function ready for another set of points. Pressing <esc> will bring up the digitize menu and display the option to quit digitizing. At this point press [L] to start digitizing a new bubble.

Get all of the bubbles that are not completely above the +10" line in one sweep. The trace will need to be slid sideways on the tablet several times to get them all. Be careful to set the bottom of tube line exactly on the A to C line each time the trace is moved.

To get the bubbles above the +10" line locate the map again with the following coordinates:

A   North=10   East=0
B   North=18   East=0
C   North=10   East=8

There is no need to digitize the X, Y, and Z points again.

Then go to the list function again but put the +10" line on the A to C line instead of the bottom of the tube line.

Large bubbles centered on the +10" line may digitize with either of these methods. In that case set the map coordinates to the following:

A   North=4   East=0
B   North=12   East=0
C   North=4   East=8

Then set the +10" line on the line 6" above the A to C line, and digitize any bubbles that were missed.

Checking Initial Data
Once the data has been stored, check the area files to see if all the diameter files are there. First, using PCTools or DOS, look at a list of the files in the trace directory, and see if every bubble is listed. If some are missing, go back into EasyDij, and digitize the major diameter for those bubbles (make sure the map is located correctly and the trace on the tablet correctly).

To check the area files, use a text editor. First, go through the file and delete any line that starts with anything but a [+] or [-]. If there are any records starting with a [-], then delete the record before that record and change the [-] to a [+]. The record before the one that starts with a [-] was selected to be replaced be the record starting with a [-](see the section on making area files).

Count the area records. If the number of area records matches the number of bubbles the data is probably good. If they do not match, then either digitize the whole trace again, or check the area of random bubbles using EasyDij until the missing or extra bubble(s) are found. Once records to delete or insert are found, the changes can be made with any ASCII text editor.

It is a good idea to check a few of the areas even if the file has the right number of records. If a bubble was skipped and another bubble was digitized twice, the file will have the right number of records. But, some or even most of the records will be shifted by one position.

To check the areas, print out the area file. Using the planimeter function of EasyDij, measure a few random bubbles. After each bubble, press <enter> instead of [+] so the planimeter menu will be displayed (the area of the last bubble which was digitized is displayed below the planimeter menu).

CORRECTING FOR OPTICAL DISTORTION

The VAR File
The VAR file contains the constants used by MAJDIA to make the corrections to the bubble data for optical distortion. See Appendix A.2 for an example of a VAR file. Only one VAR file is needed per trace, with only the first and last lines needing changes. The first line should read:

        R[refrigerant no.]#[trace no.]

If this line names a refrigerant other than the refrigerant of the current working trace, then a different VAR file needs to be copied (see Initial Setup). Change the trace number to match the number of the current working trace.

The last line is a number between 22 and 27. This is the outer diameter (in inches) of the image of the tube on the trace. (This value must be in inches; the program will convert to meters.) Replace this number with the outer diameter of the tube from the current working trace. Be careful when editing this file, the position of the numbers is critical. Do not insert or remove any lines from the file. Do not change the original VAR file in the DATA directory.

Running MAJDIA
To create a data file of all the areas and major diameters which is corrected for the optical distortion, type the following commands:
(starting from the DOS prompt)

        MAJDIA   (get into the correction program)
        D   (change data file)
        F   (change filename)

Enter the filename (B, G, or R) without any extension. MAJDIA will display the new filename. If that is the correct filename, then type:

L	(leave the filename)
V	(load VAR file)
F	(change filename)

enter the VAR filename without the extender. MAJDIA will display the filename. If that is the correct filename, then type:

L	(load the VAR file)
R	(run the correction)

MAJDIA will take a moment to read the area and major diameter files and write the DAT and MDA files. If a filename that does not exist is entered, MAJDIA will display a message to that effect. If it does, then a new filename must be chosen.

Checking Final Data
The final check that should be performed is to look at the MDA file and make sure that all of the output is in the acceptable range. The acceptable ranges are approximately:

Area	0.08 - 2.5
Major Diameter	0.1 - 2.5

If the data is off by a factor of ten or more, then there is a definite problem with the initial data file. The initial data file must be checked for missing or incorrect data (if the area and major diameter are both 0 then it is probably an initial area of 0).

If a few points are somewhat out-of-range, look at the trace to see if those bubbles were extraordinarily large, small, or elliptical. Bubbles very close to the edge of the tube can also come out larger than expected because the optical correction is greatest near the edge.

If the area seems too large, a final check can be done by comparing the area of the ellipse to that of a circle with the same major diameter. The area of the ellipse must be less than or equal to the area of a circle with the same major diameter.

$$AREA_e \leq AREA_c = \pi \left(\frac{Maj.Dia.}{2} \right)^2 \qquad (17)$$

20

Simultaneous heat transfer and visual characterization of the boiling process are good approaches to understanding the effect of bubble dynamics on a surface. Unfortunately, computer-assisted analysis of visual data becomes tedious with increasing numbers of bubbles to analyze. The use of digital image analysis permits the automation of the image analysis procedure and provides a more convenient, accurate, and consistent analysis.

The basic concepts of digital image analysis will be introduced in this section. Next the objectives, approach, and results of this method will be described. Following that, some sample analyses, along with comparisons to similar work done with the computer-assisted system, will be presented. Finally, conclusions will be drawn, and suggestions for the future direction of this work will be provided.

APPLICATION OF DIGITAL IMAGE ANALYSIS

Digital Image Analysis Background

Since digital image analysis (DIA) is a relatively new technique, introducing some basic concepts before outlining the details of this method is necessary. DIA is achieved by computer manipulation of pictures, or more specifically, images that have been converted into numeric form. Conventionally speaking, an image is a picture, photograph, or other form giving a visual representation of an object or a scene. However, it has another meaning in digital image processing: an image is a two-dimensional array of numbers. Typical image sizes are 256x256, 512x512, and 1024x1024. For example, in a 512x512 image there are 512 lines in the image and 512 numbers in each line. Each number corresponds to one small area of the visual image, indicating the level of darkness or lightness of that area. Zero represents black and an arbitrary maximum value represents white for a gray scale image. Each small area to which a number is assigned is called a "pixel" (picture element), while the number corresponding to each pixel is called the "pixel value." An example image and its corresponding pixel data are shown in Figure 7.
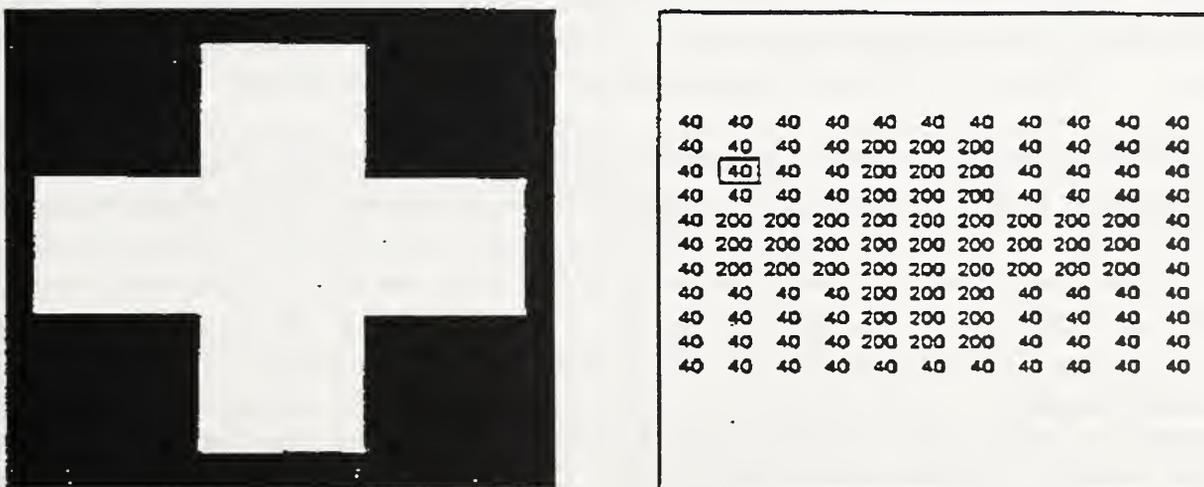
Figure 7 - A simple image and its corresponding pixel data [2]

Images can be acquired (digitized) by converting the video output from a camera to pixel data using a digital image processing board, sometimes called a "frame grabber." Images can be saved as data files or in frame buffers (memory chips on the frame grabber) and then displayed by converting each pixel value to a corresponding intensity on a monitor screen. Images are processed by manipulation of the pixel data. In general, the purpose of digital image processing is to enhance or improve images to extract useful information. Typical operations to achieve this enhancement include filtering and thresholding.

Filtering is used to reduce the noise or enhance the edges in an image. Filtering is achieved by performing a convolution of two functions, as shown in Eq. (18).

$$h(x) = \int_R f(x-\xi)g(\xi)d\xi \qquad (18)$$

Where $f(x-\xi)$ and $g(\xi)$ are called the input and the impulse response functions of the convolution, respectively, and $h(x)$ is the output function. Equation (18) is a continuous one-dimensional example, but in DIA the transform is two-dimensional and discrete, taking the form of a double summation. In that case, the impulse response is often referred to as a "kernel." (See [2] for further details.) In smoothing, also known as low-pass filtering, the impulse response can be a rectangular, triangular, or Gaussian. In edge enhancement, also known as high-pass filtering, the impulse response has a positive peak with negative side lobes. Other impulse responses provide the Laplacian and the Fourier transform. In DIA, the input function is a pixel matrix; the impulse function can be any of those described; and the output is an "enhanced" image.

Thresholding is used to segment an image into regions (e.g., objects and the background.) In thresholding, all pixels with pixel values at or above a prescribed level are assigned to an object, while those with pixel values below the prescribed level fall out of the object, or vice versa. Normally, pixels assigned to the object are given a pixel value of 0, and pixels that fall out of the object are given the highest pixel value. Further analysis, after thresholding, can then be pursued in a binary (on/off) mode, this makes the remaining DIA simpler.

DIA has been applied in medicine, cartography, industry, manufacturing, printing, and scientific research fields such as astronomy, mineralogy, fluid mechanics, particle physics, and ocean modeling. DIA has recently found application in the experimental study of boiling heat transfer, where it promises to provide a powerful tool for the efficient analysis of visual data from boiling studies.

Literature Review
In the past few years, there have been several applications of DIA in the thermal-fluid sciences. In fluid mechanics, Particle Image Velocimetry (PIV) is a rapidly growing flow visualization technique using DIA to determine the instantaneous velocity profile of a given flow field by photographically recording tracer particle and/or bubble images within the flow at discrete instances in time. Hassan et al. [3] applied DIA to obtain quantitative velocity measurements for both phases of a two-phase flow system. Huang et al. [4] applied DIA to extract velocity vectors from images of streamlined Hele-Shaw flow. Revesz [5] applied DIA and a soap-bubble

tracer technique to obtain three-dimensional velocity data in an air flow. Tzeng et al. [6] developed a laser-based, quantitative visualization system using DIA to study a high-speed, rotating-disk flow. Kobayashi et al. [7] developed a DIA system to obtain time series whole field velocity measurement of turbulent flows.

A study closely related to the research reported here was described by Bentley and Ruggles [8]. The purpose of their study was to investigate the phase distribution of an air-water flow in a narrow rectangular channel. Ink was added to water, and clear channel walls were used to allow high-speed photographs and video tape of the air-water flow field with good contrast to be taken. Images were processed to obtain information on phase distribution. The techniques used by Bentley and Ruggles are similar to the methods discussed later for single-bubble analysis; however, the approach is incapable of processing images where two or more bubbles "overlap." (We refer to this as a multiple-bubble problem.) A novel method for overcoming this difficulty will be presented later in this report.

Objectives
The goal of this project is to apply DIA techniques to the study of boiling heat transfer. Three specific tasks were identified:
- Design and construct an appropriate image digitization and analysis system for acquiring digital data from a 16 mm high-speed film format.
- Develop algorithms and software capable of recognizing single bubbles and determining their sizes and location in an image of a two-phase flow.
- Develop algorithms and software capable of recognizing multiple-bubble objects, separating them, and estimating their sizes and locations in a two-phase flow.

Closure
An introduction to the basic terminology and ideas of DIA was provided in this section. The representation of an image as an array of numbers, and the manipulation of the pixel data was described. Furthermore, the application of these methods to related problems and to the visual study of boiling heat transfer was discussed, and the objectives of this research were established. In the following section, the hardware design will be discussed.

HARDWARE DESIGN

The digital acquisition of visual data requires an integrated system capable of providing an image, viewing an image, converting an image to numeric form, and storing an image for subsequent analysis. A general approach for accomplishing this is shown schematically in Figure 8.

The DIA system consists of a film holding and positioning mechanism, an illumination box, an optical rail, a CCD camera with a zoom microscopic objective, a computer and frame grabber, and an external monitor. Component details are given in the following sections.

Film Illumination and Placement
The illumination box is designed to hold the film and illuminate it from behind. It is a 89x152x203 mm standard aluminum electrical enclosure box with an illumination window (45x45 mm) in the front, and ventilation holes in the sides. A 40 W light bulb, controlled with a rheostat, is mounted in the center of the box to provide an adjustable and stable white light source. A frosted glass plate covers the viewing window and serves as a light diffuser.
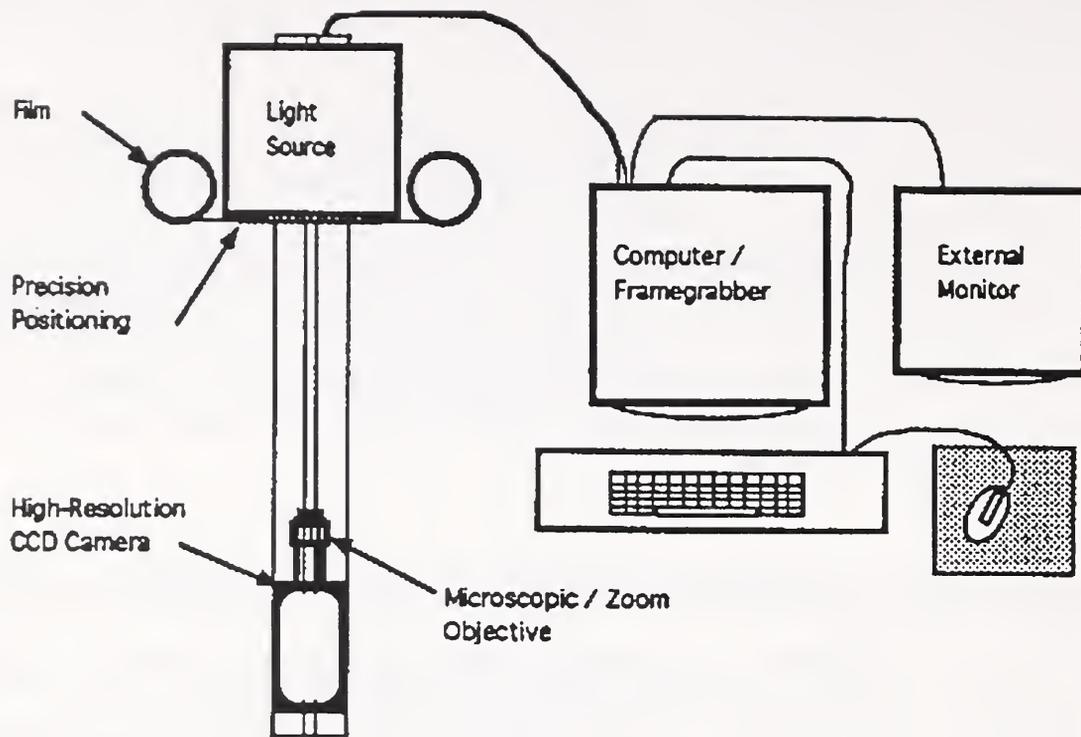
Figure 8 - A schematic of the digital image analysis apparatus

A miniature X/Y Two Axis Movement mechanism provides a two-dimensional precision traversing mechanism. The mechanism is installed in front of the illumination window, and holders consisting of small metal springs are used to secure the film, which may be precisely placed with the traversing mechanism.

Camera, Monitor and System Specifications

The camera is a high resolution CCD camera with a 640x480 pixel array. It is equipped with a micro-video-zoom objective. The camera may be positioned on a standard optical rail, upon which both the camera and illumination box are mounted. The black/white monitor has a pixel array of 640x480. The frame grabber is installed in an IBM DOS compatible computer. The frame grabber has four buffers (memory chips), and each buffer can store an image of 512x480 pixels.

System Overview and Capabilities

Visual data (high-speed motion picture images) are digitized by mounting a film frame on the traversing mechanism, illuminating it from behind with the illumination box and scanning it with the CCD camera. The use of the zoom microscopic objective, the optical rail, and the traversing mechanism allows the monitoring of video output during frame positioning and image scanning. It also provides flexibility in field-of-view and magnification during the selection and digitization of particular parts of the image. After converting an image into digital format, computer analysis is used to identify single bubbles in the two-phase flow, and to estimate their sizes, shapes, and locations. Multiple bubbles are identified, separated and their perimeters are extrapolated to completion.

24

During the early stages of the project, system performance was verified by exercising all available functions on the digital image processing board. Further testing and performance evaluations were conducted with a commercial digital image analysis package. These tests verified proper operation of the image processing board and the associated software package.

SOFTWARE DEVELOPMENT

In this chapter, the application of standard DIA methods and the new methods developed to meet the research objectives will be discussed. Many software options for single bubble analyses were explored. The three most promising of these options, DT-IRIS, Image-Pro, and NIH Image were tested with images of a two-phase flow. The capabilities of each were explored and a combination of DT-IRIS and NIH Image was selected, providing access to a wide variety of standard image analysis tools (filtering, thresholding, etc.), and allowing user definable processing. This flexibility was needed to pursue the multiple-bubble problem, which is not addressed by standard techniques, and has only recently been explored in the thermal science literature.

Single Bubble Analysis

DT-IRIS is PC-based and has two parts: DT-IRIS Tutor, and DT-IRIS subroutine libraries. DT-IRIS Tutor is a standard software package and is not modifiable, while the DT-IRIS subroutine libraries consist of FORTRAN subroutines that may be accessed or provided by the user. Both have the following capabilities:
- acquiring images from film
- saving images to data files
- filtering images
- thresholding images

NIH Image is Macintosh-based and is a non-modifiable standard software package. It has the following capabilities:
- acquiring images from film (with compatible frame grabber)
- saving images to data files
- filtering images
- thresholding images
- tracing the boundaries of bubbles manually
- measuring the sizes and locations of bubbles in hand-drawn images and images from films which have good contrast

After mounting the film on the traversing mechanism, video input from the CCD camera was digitized and stored as ".img" files using DT-IRIS Tutor. The ".img" files were translated into ".tif" files using the file translation utilities provided by Data Translation. The ".tif" files were mounted on the Macintosh for processing by NIH Image. Enhancements of the images were usually performed before analyzing. The first DIA operation applied to any image captured as described above was to correct for the distortion introduced by the pixel array difference between camera and frame buffer - the image must be stretched by a ratio of 1.25:1 (camera to frame buffer) in the horizontal direction. Then enhancement operations of sharpening, smoothing, contrast enhancement, background subtraction, and thresholding were performed. The cut-off gray level is usually decided by carefully studying the background and the object to obtain its best approximation. After enhancement, an object or dimension in the image was used

to set the scale of the image. The image was converted to a binary format, and a particle analysis was conducted. A particle analysis provides bubble size, location, shape, and statistical information.

Multiple Bubble Analysis

The physics of distinguishing multiple bubbles from a single irregularly shaped bubble relies on the well-known Young-Laplace equation describing the pressure difference across a curved bubble surface:

$$\Delta P = \sigma \kappa \qquad (19)$$

Where σ is the surface tension, and the mean surface curvature κ, is:

$$\kappa = \frac{1}{r1} + \frac{1}{r2} \qquad (20)$$

This provides a physical relationship for deciding whether the object is a single bubble or multiple overlapping bubbles. The pressure difference across a bubble surface must be finite; however, for a "cusp" or sharp corner, the pressure difference is infinite and thermodynamically impossible. This idea was used along with DIA to identify multiple bubbles. As shown in the sample analysis of the next section, this discontinuity is manifested through a dramatic change in the second derivatives of the bubble's edge curve at a cusp location. Normally, the outer edge of a digitized bubble image is not smooth; a series of piecewise least-squares curve-fits was obtained for the object. Completing the overlapped parts of each bubble presented another challenge. At the cusp location, because the signs of the first derivatives change, the first derivatives are usually small. Thus, a representative value of the first derivative (such as the first derivative at a position 10 points back from the cusp) was used for the extension. The absolute value for the second derivative was taken as the average at that point and the cusp location. With this approach, the object curve was expanded in a Taylor series about the cusp to complete the overlapping bubbles.

Identifying, separating and completing multiple bubbles is much more involved and complicated than single bubble analysis. Due to the unavailability of existing software, efforts were focused on developing this capability. DT-IRIS subroutine libraries were used because of the flexibility of user-written routines. Two FORTRAN programs have been developed.

The first program, called the Image Processing Program, utilizes DT-IRIS subroutine libraries, and is capable of:
- acquiring an image
- saving an image to a data file
- detecting the edges of multiple bubbles
- converting a data file to an image

Appendix B provides a listing and additional information about this program.

The second program (Multiple-Bubble Separation and Completion Program) is capable of:
- curve-fitting the edges of the bubbles using a least-squares method
- finding the first and second derivatives of the edge curve

26

- locating the cusps
- completing the overlapped parts of the bubbles

Appendix C provides a listing and additional information about this program.

Pixel data for images with multiple bubbles are input to the Image Processing Program which detects the edges of the bubbles and records the edge point coordinates. The Multiple-Bubble Separation and Completion Program then uses the recorded edge point coordinates as its input file and curve-fits the edges to calculate the first and second order derivatives to locate the cusp, and to complete the overlapped parts.

Examples of the procedures outlined above are presented in the next section.

SAMPLE ANALYSES

In the previous section, the application of DIA to the study of boiling and the development of new DIA techniques was discussed. In this section, these techniques will be demonstrated through their application to several images.

Sample Analyses for Single Bubbles

Five images were chosen for single bubble sample analysis. Image 1, shown in Figure 9 and Figure 10, was computer-generated. Image 2, shown in Figure 11, was taken from the original digital image analysis research proposal. These two images were chosen for analysis and comparative study to the computer-assisted system. Image 1 provides a standard (with known results), and Image 2 serves as a benchmark for comparison to the computer-assisted system. Image 3, shown in Figure 13, was taken from Bentley and Ruggles [8] and represents an early use of DIA in the thermal sciences. It was selected to demonstrate the viability of the approach when applied to images from the two-phase literature. Images 4 and 5, shown in Figure 15 and Figure 16 were taken from 16 mm films with the previous and current lighting methods, respectively. Images 4 and 5 were selected to verify the technique's usefulness in interpreting visual data from the present research and to explore the difference between the two lighting methods.

Image 1: A Computer-Drawn Image Corrected for Optical Distortion.

Image 1 is a computer-drawn image. The original and processed images are shown in Figure 9 and Figure 10, respectively. Object 1 is a circle and the other three objects are ellipses. It was assumed that the vertical height of the image corresponded to the inner radius of the test tube (arbitrarily taken as 5.0 mm before optical distortion). Table 1 compares the DIA results to those from computer-assisted system without correcting for optical distortion. The areas, as well as major and minor diameters from the DIA and computer-assisted approaches, differ by less than 3% and both methods compare favorably with the actual image content. Table 2 shows the results from both methods after correcting for optical distortion. The areas, as well as major and minor diameters, differ by less than 6%. The difference introduced by correcting for optical distortion may be due to unit conversions, computational truncation, or the pixel mapping associated with optical distortion correction. Table 3 shows the results of an energy analysis of Image 1. The energy analysis consists of estimating the energy content within each bubble, total heat load of the frame, and average energy content per bubble from the DIA results.

Table 1 - DIA results for Image 1 without correcting for optical distortion

	Area (mm^2)			Major Radius (mm)			Minor Radius (mm)		
#	Actual	DIA	Manual	Actual	DIA	Manual	Actual	DIA	Manual
1	3.142	3.150	3.160	1.000	1.000	5.993	1.000	1.000	1.012
2	1.571	1.630	1.583	1.000	1.015	1.006	0.500	0.510	0.501
3	1.571	1.630	1.583	1.000	1.010	1.020	0.500	0.510	0.494
4	1.571	1.630	1.602	1.000	1.010	1.004	0.500	0.515	0.508

Table 2 - DIA results of Image 1 after correcting for optical distortion

#	Area (mm2)		Major Diameter (mm)		Minor Diameter (mm)	
	DIA	Manual	DIA	Manual	DIA	Manual
1	2.331	2.339	2.0097	2.025	1.4804	1.471
2	1.206	1.172	1.5024	1.490	1.0200	1.002
3	1.247	1.184	2.0200	2.040	0.7850	0.739
4	1.254	1.187	1.8143	1.791	0.8776	0.844
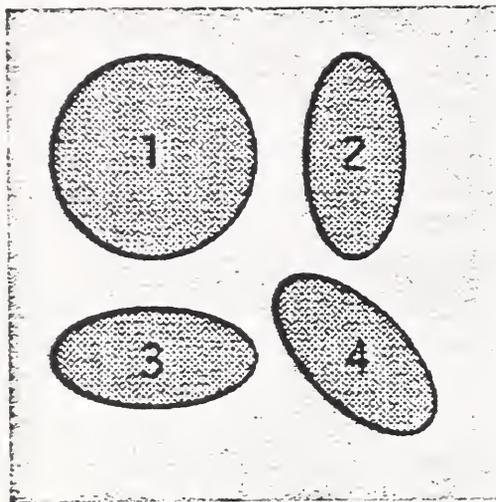
Figure 9 - Image 1: A computer-drawn image with standard objects
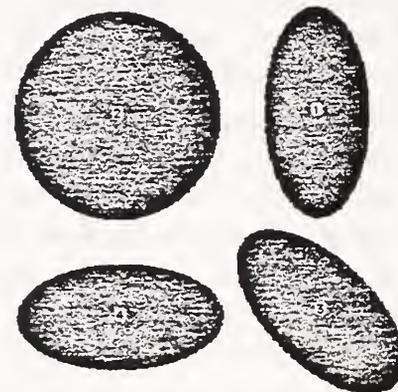
Figure 10 - Image 1 after digital image analysis

Table 3 - Energy Analysis Results of Image 1

#	Area (mm^2)	Major Diameter (mm)	Minor Diameter (mm)	Surface Area A_{el} (mm^2)	Vol. Ratio V_{sp}/V_{el}	Volume V_{el} (mm^3)	Energy Cont. Q_{el} (J)
1	2.331	2.0097	1.4804	8.57455	1.165	2.30600	0.2846E-02
2	1.206	1.0200	1.5024	7.09091	0.824	1.20542	0.1488E-02
3	1.247	2.0200	0.7850	4.13551	1.604	0.65182	0.8045E-03
4	1.254	1.8143	0.8776	4.25586	1.438	0.73168	0.9031E-03

28

Heat Load of All Bubbles

Assumed Spherical (J): 0.71312E-02

Assumed Elliptical (J): 0.60417E-02

Density of Vapor (Kg/m3): 7.29364

Latent Heat (J/Kg): 169227.00

Saturation Temperature (K): 304.13

Averaged Quantities

Cross Sectional Area (mm2): 1.50948

Major Diameter (mm): 1.71798

Minor Diameter (mm): 1.16135

Surf. Area of Ellipsoid (mm2): 6.01421

Volume of Ellipsoid (mm3): 1.22373

Required Energy to Make Ellipsoid (J/Bub.): 0.15104E-02

Where the quantities with subscript "el" were obtained by assuming a bubble is an ellipsoid rotated about it's longest axis, while those with subscript "sp" were obtained by assuming a bubble is a spheroid.

Image 2: An Image from Haley and Westwater Corrected for Optical Distortion. Image 2 was taken from Haley and Westwater [9] and was given as an example in the research proposal for this project. The original and processed images are shown in Figure 11 and Figure 12, respectively.

By assuming that the vertical height of the image corresponded to the inner radius of a test tube (arbitrarily taken as 4.5 mm before optical distortion), results from a digital image analysis could be compared to an analysis of the same image done with the computer-assisted system. Table 4 compares the DIA results with the computer-assisted system results before optical distortion correction. The areas differ by 0.15% to 6.25%, major diameters differ by 0.15% to 4.45%, while minor diameters differ by 0.0% to 7.4%. Table 5 shows results after optical distortion correction. The areas differ by 0.21% to 6.25%, major diameters differ by 0.66% to 4.46%, while minor diameters differ by 0.71% to 8.76%. Table 6 shows the energy analysis results from DIA.

It should be noted that the computer-assisted system measures areas and major diameters directly: Areas are measured using a computer digitizing tablet to trace the edges of bubbles, major diameters are measured with a scale. This process involves considerable human interaction and some judgement in the interpretation of the image. Minor diameters are calculated by assuming all bubbles are ellipsoids. DIA measures areas, major diameters, and minor diameters directly. Areas are measured by pixel counting. Major and minor diameters are measured from a best-fit ellipse. A difference in area computation is not as important as major and minor diameter measurements because volumes (and energy) are based on the diameters. Even with the area discrepancy, individual bubble volume calculations can be expected to differ by less than 10%. Since the volume calculations match closely, energy content can be accurately predicted.

Table 4 - DIA results for Image 2 without correcting for optical distortion

#	Area (mm2)		Major Radius (mm)		Minor Radius (mm)	
	DIA	Manual	DIA	Manual	DIA	Manual
1	0.650	0.649	0.645	0.646	0.320	0.320
2	0.530	0.552	0.565	0.586	0.300	0.300
3	0.800	0.750	0.700	0.707	0.365	0.338
4	0.510	0.525	0.430	0.440	0.375	0.380
5	0.360	0.383	0.400	0.415	0.285	0.294
6	0.910	0.969	0.730	0.764	0.395	0.400
7	0.360	0.373	0.405	0.398	0.285	0.299

Figure 11 - Image 2: An image from Haley and Westwater after digitization [9]

Figure 12 - Image 2 after digital image analysis

Table 5 - DIA results for Image 2 after correcting for optical distortion

	Area (mm2)		Major Diameter (mm)		Minor Diameter (mm)	
#	DIA	Manual	DIA	Manual	DIA	Manual
1	0.482	0.481	1.2422	1.2340	0.4925	0.4960
2	0.393	0.409	1.1300	1.1720	0.4446	0.4440
3	0.592	0.555	1.3165	1.3460	0.5743	0.5240
4	0.377	0.389	0.8562	0.8800	0.5572	0.5620
5	0.267	0.284	0.7998	0.8300	0.4235	0.4360
6	0.681	0.718	1.4599	1.5280	0.5911	0.5980
7	0.285	0.296	0.8081	0.7960	0.4521	0.4720

Table 6 - Energy analysis results for Image 2

#	Area (mm^2)	Major Diameter (mm)	Minor Diameter (mm)	Surface Area A_{el} (mm^2)	Vol. Ratio V_{sp}/V_{el}	Volume V_{el} (mm^3)	Energy Cont. Q_{el} (J)
1	0.482	1.2422	0.4925	1.59854	1.588	0.15777	0.1947E-03
2	0.393	1.1300	0.4446	1.31156	1.594	0.11693	0.1443E-03
3	0.592	1.3165	0.5743	1.99559	1.514	0.22738	0.2806E-03
4	0.377	0.8562	0.5572	1.33860	1.240	0.13918	0.1718E-03
5	0.267	0.7998	0.4235	0.91689	1.374	0.07510	0.9270E-04
6	0.681	1.4599	0.5911	2.25946	1.572	0.26707	0.3296E-03
7	0.285	0.8081	0.4521	0.99758	1.337	0.08649	0.1068E-03

Heat Load of All Bubbles
   Assumed Spherical (J): 0.19653E-02
   Assumed Elliptical (J): 0.13206E-02
   Density of Vapor (Kg/m3): 7.29364
   Latent Heat (J/Kg): 169227.00
   Saturation Temperature (K): 304.13

Averaged Quantities
   Cross Sectional Area (mm2): 0.43950
   Major Diameter (mm): 1.08751
   Minor Diameter (mm): 0.50504
   Surf. Area of Ellipsoid (mm2): 1.48832
   Volume of Ellipsoid (mm3): 0.15284
       Required Energy to Make Ellipsoid (J/Bub.): 0.18865E-03

     Image 3: An Image from Bentley and Ruggles. Image 3 is an image of the phase distribution of air-water flow in a narrow rectangular channel [8]. Ink was added to the water to provide satisfactory contrast between the bubbles and the water. A clear channel was used. Figure 13 and Figure 14 are the original and processed images, respectively. Table 7 lists the dimensions of the bubbles by assuming the horizontal frame length to be 10 cm.
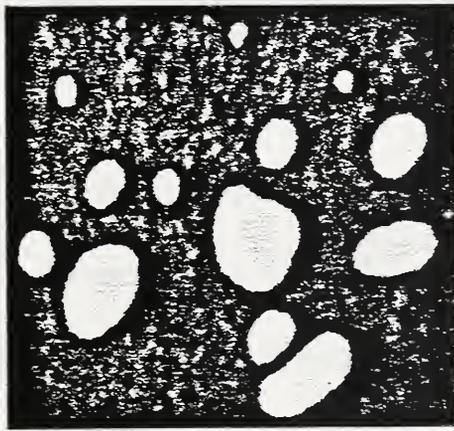
Figure 13 - Image 3: An image from Bentley and Ruggles [8]

Figure 14 - Image 3 after digital image analysis

Table 7 - Direct measurement of Image 3

#	Area (cm2)	Maj. Diam. (cm)	Min. Diam. (cm)	Angle (degree)
1	0.08	0.37	0.28	73.79
2	0.07	0.32	0.27	98.17
3	0.17	0.54	0.40	95.21
4	1.09	1.37	1.01	63.43
5	0.62	1.01	0.79	68.02
6	0.59	1.00	0.75	61.88
7	0.26	0.68	0.48	90.11
8	3.08	2.18	1.80	109.86
9	1.47	1.77	1.06	16.02
10	0.56	0.92	0.77	93.17
11	2.43	2.14	1.45	59.23
12	0.75	1.16	0.82	53.04
13	2.08	2.37	1.12	38.72

Image 4: Bubbles from 16 mm Film with Back Lighting. Image 4 is from a 16 mm film with an early lighting method. The image was recorded during boiling, with a light source behind the test tube. The original and processed images are shown in Figure 15. The analytical results given in Table 8 demonstrate the application of this technique to 16 mm films.
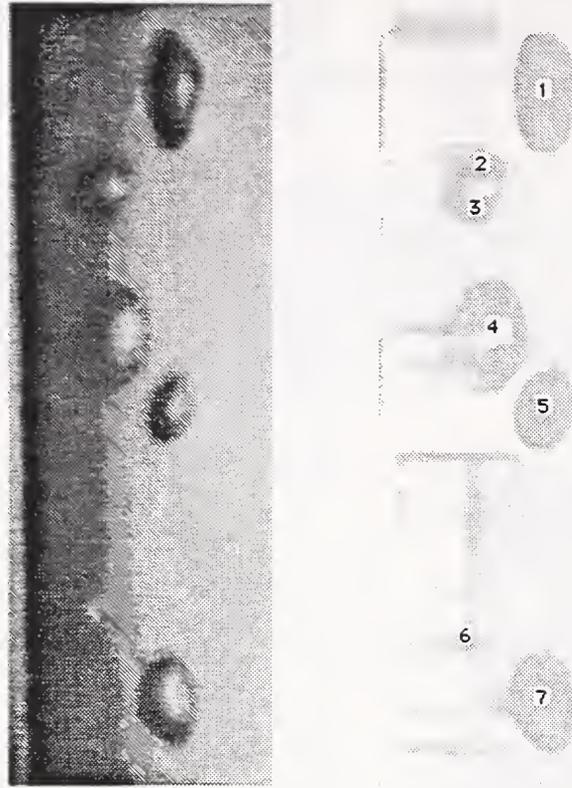
Figure 15 - Image 4: Original and processed images from 16mm film with back lighting

Table 8 - Results from a digital image analysis of Image 4

#	Area mm^2	x-location mm	y-location mm	Volume mm^3
1	130.33	27.67	116.00	1043.21
2	20.11	17.33	103.66	52.94
3	11.33	16.33	96.33	22.45
4	98.10	19.33	76.33	549.37
5	82.55	27.67	63.00	572.57
6	6.89	14.67	24.67	10.95
7	97.22	27.33	14.33	842.49

Image 5: Bubbles from 16 mm Film with Current Lighting Method. Image 5 is also from 16 mm film, but with the current lighting method. In the current lighting method, the test tube is illuminated from both behind and above. The original and processed images are shown in Figure 16, and results from the analysis are given in Table 9. The analysis demonstrates the flexibility of the technique through its ability to handle both lighting methods.

32
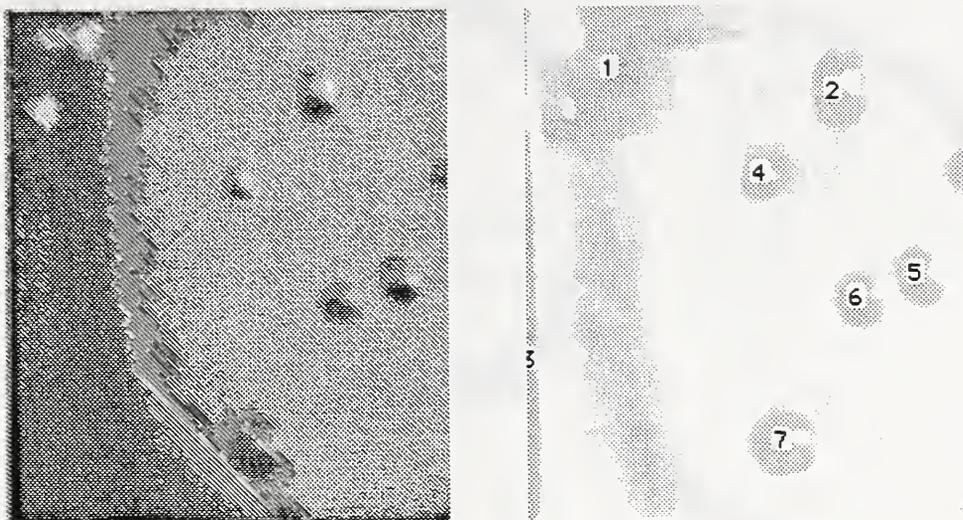
Figure 16 - Image 5: Original and processed images from NIST film with current lighting

Table 9 - Results from a digital image analysis of Image 5

#	Area (mm^2)	x-location (mm)	y-location (mm)	Volume (mm^3)
1	236.28	12.66	62.30	2059.48
2	36.18	42.64	58.63	120.78
3	52.38	0.33	21.65	44.02
4	21.75	32.65	47.64	64.76
5	29.19	53.97	33.31	92.35
6	23.31	45.64	29.98	73.10
7	41.73	34.98	10.33	179.27

Sample Analysis for Multiple Bubbles

Two images have been chosen for multiple bubble analysis. The first image is a hand-drawn image and the second image is taken from 16mm film.

Image 6: A Hand-Drawn Image with Multiple Bubbles. Shown in Figure 17 is a hand-drawn image of two overlapping bubbles after thresholding. The top and bottom edges and edge-point coordinates were detected and recorded using the Image Processing Program (Appendix B). The edge point coordinates were then used by the Multiple Bubble Processing Program to obtain derivative information, determine cusp locations, and complete the overlapping objects. The first and second derivatives of the bubble edge are shown in Figure 18 and Figure 19, respectively. The second derivative increases sharply at each cusp location. This signature in the second derivatives at each cusp location verifies our prediction from Young-Laplace relationship between pressure, surface tension, and surface curvature. A bubble

33

Figure 17 - Image 6: A hand-drawn image with multiple bubbles
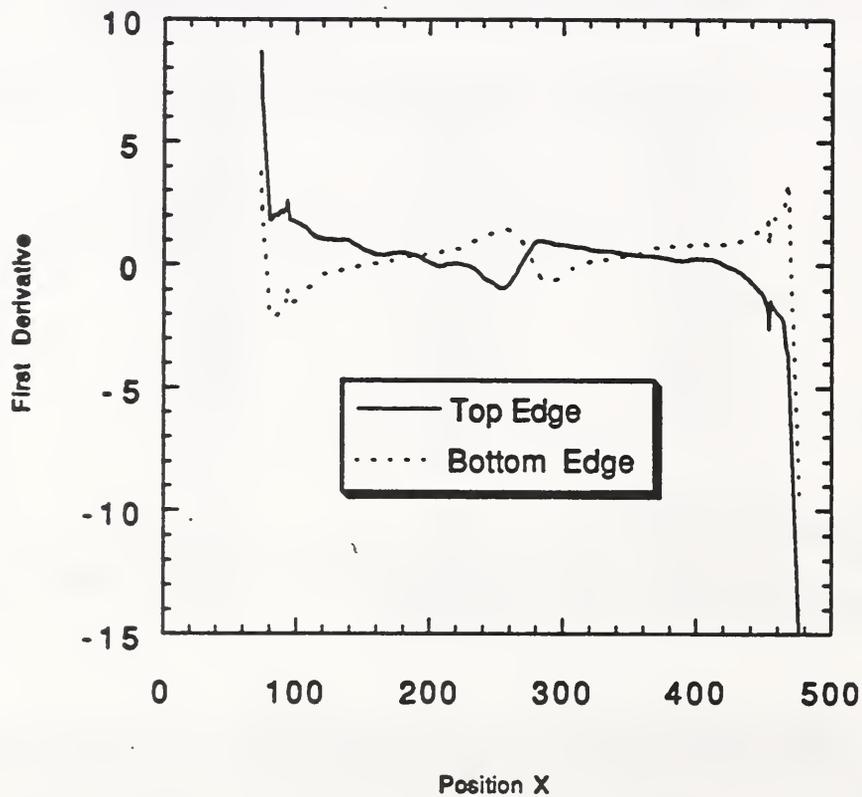
Figure 18 - The first derivatives of the edge-point curve of Image 6

surface cannot exhibit a cusp. The pressure difference across the bubble surface must be finite, and a cusp is an impossible thermodynamic singularity. Thus, we can identify the cusp location. This simple technique has been highly successful with the multiple-bubble images analyzed to date. A small box is drawn around the cusp location obtained: The success of the method is
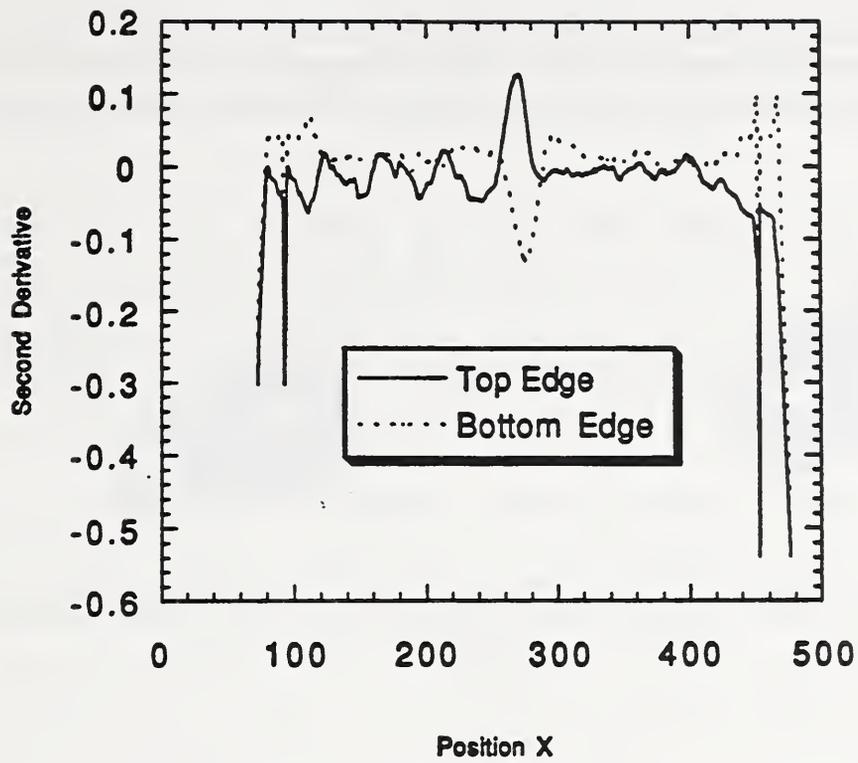
Figure 19 - The second derivatives of the edge-point curve of Image 6

demonstrated in Figure 20. The dash lines in Figure 20 show the extended edges of the bubbles from a Taylor series expansion from the left or right about the cusp.
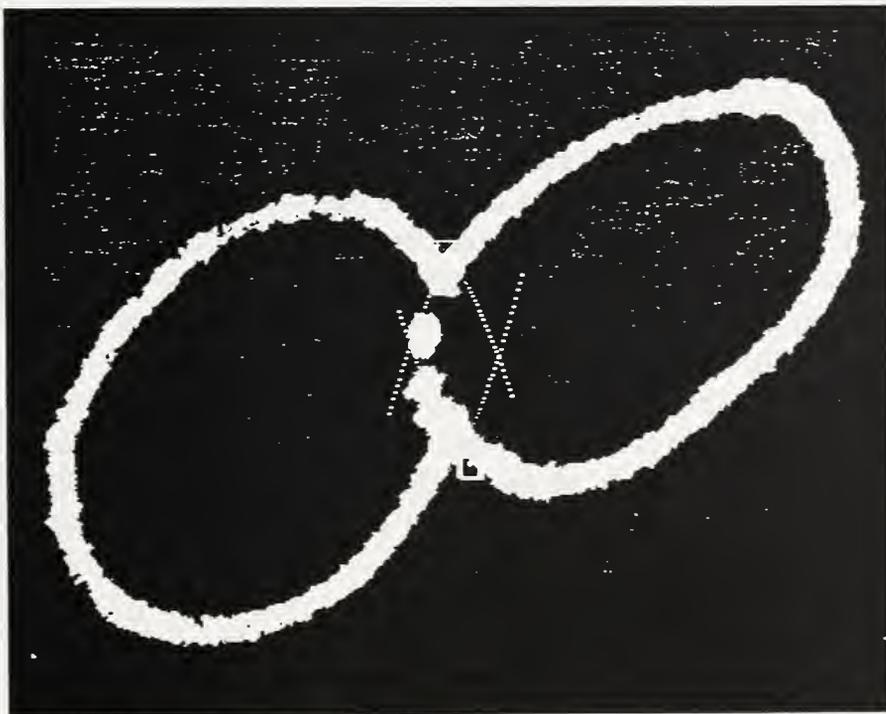
Figure 20 - Image 6 after digital image analysis

35

Image 7: An Image with Multiple Bubbles from 16 mm film. Image 7 is an image of multiple bubbles taken from a 16 mm film. The image is shown after DIA (with thresholding) in Figure 21. Following the same routine used for Image 6, we obtained the processed image as shown in Figure 22.

Figure 21 - Image 7: An image with multiple bubbles from a NIST film

Figure 22 - Image 7 after digital image analysis

Summary
The results presented in this section demonstrate very good agreement between the DIA methods developed and the computer-assisted system for visual measurements. Furthermore, both techniques were verified through comparison to a standard image. Images from the literature and from 16 mm films were successfully analyzed, and energy content data were extracted using DIA techniques. An innovative approach for separating and completing multiple-bubble objects was explained and demonstrated.

Overall, these methods are a success but some problems have been encountered. The contrast (bubble to background) of the 16 mm films often requires human interaction in application of the DIA techniques. This is partially due to, or exacerbated by, image noise (bright spots and shadows) and illumination inconsistency. The efficient use of the DIA methods presented may require lighting changes to improve image contrast. On the other hand, preliminary work on time-sequence analysis of the bubble images indicates that statistical methods may help overcome some of the difficulties associated with poor image quality. Another problem, this one due to the DIA approach, has been encountered when trying to complete some multiple-bubble objects. For example, in the analysis of Image 5, about 40% of the bubble on the right is hidden, the cusp is successfully located but object completion is obviously incorrect. The incorrectness is due to the shape of this particular bubble and is not always encountered. The population for this type of bubbles is thought to be relatively small compared to that of single bubbles. A heuristically determined correction (small) for this type of multiple bubble can be easily implemented.

CONCLUSIONS

In this report, the motivation and background for this research were discussed and the following tasks identified:
- Design and construct an appropriate image digitization and analysis system for acquiring digital data from a 16 mm high-speed film format.
- Develop algorithms and software capable of recognizing single bubbles and

determining their sizes, shapes and locations in a two-phase flow.

- Develop algorithms and software capable of recognizing multiple-bubble objects, separating them, and estimating their sizes and locations in a two-phase flow.

The detection of single bubbles in an image (many single but not overlapping bubbles) was discussed, and several example analyses were presented. The DIA methods involve enhancement, thresholding and particle analysis, and post processing. The techniques and computer programs implementing them were explained and demonstrated. Comparisons to a standard image and to the current computer-assisted system verified the validity of the approach when applied to images taken from the literature and 16 mm films.

To identify and separate overlapping bubbles (multiple bubbles), a novel DIA approach exploiting the physics was explained, and examples were provided. The approach relies on a surface curvature signature (a characteristic peak in the second derivative) for cusp location, and on a modified Taylor series expansion for bubble completion. The computer implementation of this method was detailed, and it was applied to multiple bubbles that were hand drawn or obtained from 16 mm film. Cusp location is highly reliable; unfortunately, the bubble completion scheme was noted to fail for some (small) number of bubbles.

CONCLUSION AND FUTURE POSSIBILITIES

The current approach to DIA seems to have reached its limits. Because of limitations imposed by the contrast of the film, overlapping bubbles, and PC computing power, this approach will probably not result in a usable method for studying bubbles.

RESULTS TO DATE

Currently, the computer-assisted method of taking data is the best available method. The DIA method seems to promise the solutions to our problems; however, it still needs a lot of work before it will be able to replace the computer-assisted system.

NISTIR 4948
For the NIST Internal Report 4948 "Simultaneous Visual and Calorimetric Measurements of R11, R123, and R123/Alkylbenzene Nucleate Flow Boiling" all of the visualization data was taken using the computer-assisted method. The accuracy of the volume calculation for two confidence intervals was estimated to be 22% of the measured value.

COMPARISON OF DIA TO COMPUTER-ASSISTED SYSTEM

The two methods can be compared only for relatively contrived circumstances[6] because the DIA method can only handle bubbles that slightly overlap. However, in these cases, the two methods match to within 7.5% before optical correction, and to within 8.8% after optical correction. This is a good indication that if the DIA method becomes effective for overlapping bubbles as well as for independent bubbles, it will easily supersede the current method.

The DIA method would also significantly improve the amount of data that can be collected. The DIA method should allow us to take data from more frames with less human attention. This will, in turn, increase the accuracy and reliability of our results by reducing the error inherent in using a small set of data.

FUTURE DIA POSSIBILITIES

The digitization system design presented in this report is adequate for single-frame studies; however, improvements are possible and needed if image sequences are to be studied in the future. A Macintosh compatible frame grabber should be used and substitutes for the DT-IRIS routines found or developed. This would allow all DIA to be performed on a single platform and would make the process easier and more efficient. The holding and illumination of film should be improved so that frame positioning and illumination are consistent and repeatable for sequence digitization.

Although pixel-area-to-bubble volume mapping has not been examined thoroughly in this project, future work should explore the implications of the major-axis ellipsoidal model. Rotating about the major axis introduces a downward bias in volume calculation, and this concern should be

---

[6] See the analysis of image 1 in the section on DIA.

resolved.

Single and multiple bubble analyses should be written in the image macro language to alleviate the need for human interaction. A machine-driven analysis is only possible if the NIST film contrast is improved and consistent — and if illumination box performance is repeatable.

The improvements outlined above would allow automated DIA, and provide the tools needed to study the relation between bubble population density and distribution, bubble shapes, sizes, and interfacial characteristics, and the boiling heat transfer performance of alternative refrigerants.

Finally, by moving all the analysis to a single platform and undertaking the modest digitization changes outlined above, sequences of images could be analyzed for transient and rate behavior. This would be pursued through auto- and cross-correlation of image data to 'pair' and track bubbles as they grow, detach, and join the two-phase flow. DIA would then be capable of providing an extended quantitative analysis of the role of bubble dynamics in the boiling performance of alternative refrigerants and refrigerant/oil mixtures.

By shifting the approach, the route to successful digital image analysis may be found. Some of the possibilities that have been discussed are:

- A vector approach on a workstation or mainframe computer.
- A heuristic approach, approximating the overlapping bubbles by what other similar bubbles look like.
- Approximating the shape of partially hidden bubbles with an ellipse.
- Performing time sequence analysis of individual bubbles. The shape of the bubble in the frames before and after may help show the shape of the bubble in the current frame.

REFERENCES

[1] Kedzierski, M. A., 1992, "Simultaneous Visual and Calorimetric Measurements of R11, R123, and R123/Alkylbenzene Nucleate Flow Boiling," NISTIR 4948, National Institute of Standards and Technology, Gaithersburg, Maryland.

[2] Niblack, Wayne, 1986, "An Introduction to Digital Image Processing," Prentice/Hall International, Reading.

[3] Hassan, Y. A., Canaan, R. E., Blanchat, T. K., and Seeley, C. H., "Simultaneous Velocity Measurements of Both Components of a Two Phase Flow Using Particle Image Velocimetry," FED-Vol. 128, pp85-99, Experimental and Numerical Flow Visualization, ASME 1991.

[4] Huang, J., Narasimhan, R., and Lee, S. S., "Quantitative Analysis of Streamline Flow Using Image Processing," FED-Vol. 128, pp101-107, Experimental and Numerical Flow Visualization, ASME 1991.

[5] Revesz, Z., "Quantification of 3-D Velocity Distribution in Air Flow Using Soap-Bubble Tracers, Color Photography and Digital Image Processing," FED-Vol.128, pp109-112, Experimental and Numerical Flow Visualization, ASME 1991.

[6] Tzeng, H-M., Munce, A. C., and Crawforth, L., "Quantitative Air Flow Visualization Between Shrouded Corotating Disks," FED-Vol.128, pp141-147, Experimental and Numerical Flow Visualization, ASME 1991.

[7] Kobayashi, K., Sakakibara, J., Hishida, K., and Maeda, M., "Time-Series Measurements of Turbulent Flow Field Using Image Processing System", FED-Vol.128, pp155-162, Experimental and Numerical Flow Visualization, ASME 1991.

[8] Bentley, C, and Ruggles, A., "Phase Distribution Measurements in Narrow Rectangular Channels Using Image Processing Techniques," FED-Vol.125, pp9-15, Experimental Techniques in Multiphase Flows, ASME 1991.

[9] Haley, K. W., and Westwater, J. W., "Heat Transfer from a Fin to a Boiling Liquid'," Chem. Engr. Sci., Vol. 20, p. 711, 1965

[10] Sweeney, R.J., 1953, "Measurement Techniques in Mechanical Engineering," John Wiley and Sons, pp27-29

[11] McClain, M., 1992, Personal Communications, National Institute of Standards and Technology.

APPENDIX A.1 - FORTRAN CODE FOR MAJDIA.FOR

```fortran
        PROGRAM MAJDIA
c ********************************************
c *       JOE CROWDER
c *   Find the major diameter of the bubble
c *   given the list of points on the circumference.
c *   Check the area using triangles to see if Maj.
c *   Dia. splits the area.
c *
c *   Started 4/1/92   (April Fool's Day)
c *
c *   2/10/93 - fixed angle calculation in CORRECT - J.C.
c *
c ********************************************
C *
C *   IMPORTANT VARIABLES:
C *   FILE - NAME OF INPUT AND OUTPUT FILES AND DIRECTORY (B, G, R)
C *   VARFILE - NAME OF FILE AND DIRECTORY CONTAINING DISTORTION
C *       CONSTANTS
C *   DIR - DIRECTORY NAME
C *   SELECT - MAIN MENU SELECTION
C *   K,L,M,R,S,T,U,V,LIM,OD - CONSTANTS IN THE CORRECTION ROUTINE
C *   VARBLANK - WHETHER OR NOT A VAR FILE HAS BEEN READ
C *   OUTFILE - NAME OF OUTPUT FILE (B.MDA, R.MDA, G.MDA)
C *   AREAFILE - NAME OF AREA FILE(B.ARE, G.ARE, R.ARE)
C *   L(1000,3) - UP TO 1000 POINTS ON THE CIRCUMFERENCE OF THE BUBBLE,
C *           2 AND 3 ARE THE EAST AND NORTH POSITIONS RESPECTIVELY
C *   MAX - MAXIMUM DISTANCE BETWEEN TWO POINTS ON THE
C *       CIRCUMFERENCE
C *   MAJ - MAJOR DIAMETER OF CURRENT BUBBLE
C *   AREA - AREA OF CURRENT BUBBLE
C *   VERT - VERTICAL POSITION OF CURRENT BUBBLE
C *   ANGLE - ANGLE FROM VERTICAL OF THE MAJOR DIAMETER
C *
C *
C ********************************************
        implicit real*8 (a-z)
        character dir*22, fil*8, file*34, select*1, varfile*34
        real k,l,m,r,s,t,u,v,lim,od
        logical varblank
        common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank

        dir=' '
        fil='1B'
        file='1B'
        varfile='none'
        select='g'
        varblank=.true.
c------MAIN MENU
```

```fortran
         do 50 while ((select.ne.'q').and.(select.ne.'Q'))
            len=nblank(file)
7        write (*,5) file(1:len),varfile,file(1:len)
            write (*,*)
            write (*,*)
            write (*,10)
            read (*,20, err=7) select
5        format (/////13x,'Input Files:  ',a,'.001 - .999 and .ARE'
     *      /10x,'Variables File:  ',a/13x,'Output File:  ',a,'.MDA')
10       format (5x,'Change D)ata File Set'/5x,
     *      'Load A V)ariable File'/5x,
c    *       'M)ake A Variable Set'/5x,
     *      'R)un (this creates a new file with the .MDA extender'/5x,
     *      '       and a .DAT file in F:\EASYDIJ\DATA\)'
     *      /5x,'Q)uit to DOS'/5x,
     *      /10x,'selection:')
20       format (a)
            if (select.eq.'d') call change_name (dir, fil, file)
            if (select.eq.'v') call loadvar (dir, fil, file, varfile)
            if (select.eq.'m') call makevar (varfile)
            if (select.eq.'s') call showvar (varfile)
            if (select.eq.'r') call loop (dir, fil, file)
50    continue
         STOP
         END




         SUBROUTINE change_name (dir, fil, file)
c-----------------------------------------------------------------
c------CHANGE DATA FILENAME
         implicit real*8 (a-z)
         character   s*1, file*34, dir*22, fil*8, file2*34
2     write (*,5) file
         read (*,*) s
5     format (///10x,a,/5x,'Do you wish to change F)ilename or ',
     *'D)irectory'/5x, 'or L)eave it?'//10x,'choice: ')
         if ((s.eq.'d').or.(s.eq.'D')) then
            write (*,10) 'directory', dir, 'directory'
            read (*,20) dir
         end if
         if ((s.eq.'f').or.(s.eq.'F')) then
            write (*,10) 'file', fil, 'file'
            read (*,20) fil
         end if
10    format (/5x,'Old ',a,' name:',a/5x,'New ',a,' name:')
20    format (a)
         file=dir
         file(nblank(dir)+1:34)=fil
         if (s.ne.'l') goto 2
         file2=file
```

```fortran
      file2(nblank(file)+1:30)='.are'
      open (5, file=file2, status='old', err=999, blank='null')
      close (5)
      RETURN
999   write (*,1000)
1000  format (5x,'FILE DOES NOT EXIST')
      RETURN
      END


      SUBROUTINE loadvar (dir, fil, file, varfile)
c-------------------------------------------------------------------
c------LOAD VARIABLES FOR CORRECTION FORMULAE
c------UNIT=2 IS THE VARIABLES FILE
      implicit real*8 (a-z)
      character film*7
      common /f/film
      real k,l,m,r,s,t,u,v,lim,od
      logical varblank
      common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank
      character file*34, varfile*34, dir*22, fil*8,
     * varfil*8, sel*1

      if ((varfile.eq.'manual') .or. (varfile.eq.'none')) then
         varfile=file
         varfile(nblank(file)+1:34)='.VAR'
      end if
      varfil=fil


c------OPT FOR DIFFERENT VAR FILE
2  ·  write (*,5) varfile
      read (*,*) sel
5     format (///10x,a,/5x,'Do you wish to change F)ilename or',
     *' L)eave it?'/10x,'choice: ')
      if ((sel.eq.'f').or.(sel.eq.'F')) then
          write (*,10) 'variable file', varfil, 'variable file'
          read (*,20) varfil
          varfile=dir
          varfile(nblank(dir)+1:34)=varfil
          varfile(nblank(varfile)+1:34)='.VAR'
      end if
10    format (/5x,'Old ',a,' name:',a/5x,'New ',a,' name:')
20    format (a)
      if (sel.ne.'l') goto 2
c------GET VARIABLES
      open (2, file=varfile, status='old', err=999, blank='null')
      read (2,*) film
      read (2,*) k,l,m,r,s,t,u,v
      read (2,*) lim,od
      close (2)
      varblank = .false.
```

43

```fortran
        RETURN
999     write (*,1000)
1000    format(5x,'FILE DOES NOT EXIST'/5X,'NO VARIABLES READ, TRY AGAIN')
        varblank=.true.
        varfile='none'
        close (2)
        RETURN
        END

        SUBROUTINE makevar (varfile)
c----------------------------------------------------------------
c------CHOOSE NEW VARIABLES
c------UNIT=2 IS THE VARIABLE FILE
        implicit real*8 (a-z)
        character varfile*34
        real k,l,m,r,s,t,u,v,lim,od
        logical varblank
        common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank
        write (*,*) 'K='
        read (*,*) k
        write (*,*) 'L='
        read (*,*) l
        write (*,*) 'M='
        read (*,*) m
        write (*,*) 'R='
        read (*,*) r
        write (*,*) 'S='
        read (*,*) s
        write (*,*) 'T='
        read (*,*) t
        write (*,*) 'U='
        read (*,*) u
        write (*,*) 'V='
        read (*,*) v
        write (*,*) 'LIM='
        read (*,*) lim
        write (*,*) 'OD='
        read (*,*) od

        varblank=.false.
        varfile='manual'
        RETURN
        END

        SUBROUTINE showvar(varfile)
c----------------------------------------------------------------
        implicit real*8 (a-z)
        character touch*1, varfile*34
        real k,l,m,r,s,t,u,v,lim,od
        logical varblank
```

```fortran
        common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank

        write (*,*) 'Variable File:  ',varfile
        write (*,10) k,l,m,r,s,t,u,v,lim,od
10      format(5x,1p,'K=''g12.4/5x,'L='g12.4/5x,'M='g12.4/5x,'R='g12.4/5x,
      * 'S='g12.4/5x,'T='g12.4/5x,'U='g12.4/5x,'V='g12.4//5x,
      * 'LIM='g12.4/5x,'OD='g12.4)
        read (*,*) touch
        write (*,*) touch
        RETURN
        END

        SUBROUTINE loop (dir, fil, infile)
c----------------------------------------------------------------
c------unit=4 is the list of areas of the bubbles
c------unit=5 is a major diameter file
c------unit=6 is the output data file (i.e. ready for VISDAT)

        implicit real*8 (a-z)
        character infile*30, outfile*34, areafile*34, num*4
        logical outexist, done
        integer count
        real*8 maj
        real k,l,m,r,s,t,u,v,lim,od
        logical varblank
        common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank

c------CHECK VARS AND RESULTS FILE
c       open (7,'expdat.dat',position='rewind')

c------CHECK IF VARIABLES ARE LOADED
        if (varblank) call makevar ()

c------OPEN AREA AND OUTPUT FILES
        outfile=infile
        outfile(nblank(infile)+1:30)='.MDA'
        areafile=infile
        areafile(nblank(infile)+1:30)='.ARE'
        done=.false.
c
        goto 15
12      call confirm (dir, fil, outfile)
15      inquire (file=outfile, exist=outexist)
        if (outexist) goto 12
c
        open (6, file=outfile , status='new', err=1500)
        close (6)
        open (6,file=outfile,status='old',access='append',err=2000)
        open (4, file=areafile, status='old', err=2500)
```

```fortran
c------TITLE LINES FOR OUTPUT FILE
      write (6,*) outfile
      write (6,*) '       area            maj. dia'
      read (4,30)
30    format (/////)

c------READ INPUT FILE AND CALCULATE AREA AND MAJ. DIA.
      count=0
      num(1:1)='.'
      do 750 hun=48,57
      do 750 ten=48,57
      do 750 one=49,58
         count=count+1
         if (one.eq.58) then
           ones=48
           tens=ten+1
           else
           ones=one
           tens=ten
         end if
         if (tens.eq.58) then
           tens=48
           huns=hun+1
           else
           huns=hun
         end if
         num (2:2)=char(huns)
         num (3:3)=char(tens)
         num (4:4)=char(ones)
         write (*,*) 'Bubble #',num(2:4)
         call readfile (infile,num,area,maj,done)
         if (done) goto 760
         write (6,*) num(2:4), area, maj
750   continue
760   write (*,740) infile,count-1
      close (7)
      close (4)
      call mark (fil, count-1)
      close (6)
c
740   format (//5x,'The last file accessed was ',a,' #',i3)
      RETURN
1500  write (*,1000) 'opening output file'
      RETURN
2000  write (*,1000) 'writing to output file'
      goto 760
2500  write (*,1000) 'finding area file'
1000  format (5x,'FILE ERROR:    ',a)
      RETURN
      END
```

```
      SUBROUTINE confirm (dir, fil, file)
c-----------------------------------------------------------------
c------CHOOSE WHETHER TO OVERWRITE OLD OUTPUT FILE

      implicit real*8 (a-z)
      character file*34, choice*1
      write (*,50) file
      read (*,*) choice
50    format (5x,'The output file for ',a,' already exists.'/5x,
     * 'Do you wish to R)eplace it or C)hoose a different set of files',
     * //10x,'   Choice: ')
      if (choice.ne.'r') then
          call change_name (dir, fil, file)
          goto 100
      end if
      open (6,file, status='old')
      close (6,status='delete')
100   RETURN
      END


      SUBROUTINE readfile (file,num,area,maj,done)
c-----------------------------------------------------------------
c------unit=4 is the list of areas of the bubbles
c------unit=5 is a major diameter file
c------unit=6 is the output data file (i.e. ready for VISDAT)

      implicit real*8 (a-z)
      real*8 maj, l(1000,3)
      character file*34, file2*34, num*4
      logical done
      done=.false.
      do 5 i=1,1000
          do 5 j=1,3
             l(i,j)=0
5     continue

C------READ POINTS ON CIRCUMFERENCE FROM FILE
      file2=file
      file2(nblank(file)+1:34)=num
      open (5, file=file2, status='old', err=999, blank='null')
      read (5,10)
10    format (///)
      do 30 i=1,1000
          read (5,*, end=500) l(i,1),l(i,2),l(i,3)
          write (*,*)l(i,1),l(i,2),l(i,3)
30    continue
500   close (5)
      no=i-1
      read (4,80) area
80    format (/34x,f6.3)
```

47

```fortran
      if (area.eq.0) read (4,90) area
90    format (34x,f6.3)
C------CALCULATE APPARENT MAJ. DIA. AND ANGLE, THEN CALCULATE
C------ACTUAL MAJ. DIA. AND AREA
      call findmax (no,l,maj,angle,vert)
      call print_out (vert, area, maj, angle,'Original      ')
      call correct (area,maj,angle,vert)

      call print_out (vert, area, maj, angle,'Corrected     ')

      RETURN
999   done=.true.
1000  format (5x,'FILE ERROR:    ',a)
      END




      SUBROUTINE findmax (no,l,max,angle,vert)
c----------------------------------------------------------------------
c------FIND THE MAJOR DIAMETER OF THE BUBBLE. l IS THE LIST OF
COORDINATES.

      implicit real*8 (a-z)
      real*8 l(1000,3), m, max
      integer i, j
      max=0.d0
50    format (x,i3,' points on circumference of ellipse')
C------FIND THE MAXIMUM DISTANCE BETWEEN TWO POINTS ON THE
CIRCUMFERENCE
      do 1000 i=1,no-1
         do 500 j=i+1,no
           m=sqrt((l(i,2)-l(j,2))**2+(l(i,3)-l(j,3))**2)
           if (m .GT. max) then
              max=m
              pos1=i
              pos2=j
           end if
500      continue
1000  continue
C------FIND THE APPARENT ANGLE AND THE VERTICAL POSITION
      x1=l(pos1,3)
      y1=l(pos1,2)
      x2=l(pos2,3)
      y2=l(pos2,2)
      if (x2.eq.x1) then
          angle=3.1415926535D0/2.D0
      else
          angle=abs(atan((y2-y1)/(x2-x1)))
      end if
      vert=(y2+y1)/2.D0
```

48

```fortran
      if (vert.le.0.5) then
          vert=0.5D0
      end if

      RETURN
      END

      SUBROUTINE correct (area,maj,angle,vert)
c----------------------------------------------------------------
c------correct for the optical distortion and change of units
c------hfht = > correct trace measurements (in) to apparent image meas.(mm)
c------imarea = > area of bubble image (sq mm)
c------rmaj = > major radius of apparent ellipse (mm)
c------rmin = > minor radius of apparent ellipse (mm)
c------angle = > angle of apparent ellipse (rad)
c------dc = > distance from center of tube to center of bubble (mm)
c------dst = > vertical distortion factor (mm/mm) depends on dc
c------A, B, C = > from definition of ellipse
c------rmaja, rmina = > major and minor radii of actual ellipse
c------ang = > angle of actual ellipse (rad)
      implicit real*8 (a-z)
      real*8 hfht, imarea, dc, rmaj, rmin, maj, dst
      real k,l,m,r,s,t,u,v,lim,od
      logical varblank
      common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank


c------CONVERT INCHES ON TRACE TO APPARENT MILIMETERS
      hfht=0.451D0/od*25.4D0
      imarea=area*hfht**2
      dc=abs((od/2.0D0-vert)*hfht)
      rmaj=maj/2.0D0*hfht
      rmin=imarea/3.1415926535D0/rmaj

      write (*,*) 'OD=',od
      write (*,*) 'hf/ht=',hfht,'    image area=',imarea
      write (*,*) 'dist from cen.=',dc,'    maj./min. axes',rmaj,rmin

c------CORRECT FOR VERTICAL DISTORTION r=distortion factor
      if (dc.le.lim) then
          dst=k*dc**2+l*dc+m
      else
          dst=r*exp(s*dc+t)+u*dc+v
      end if
c
      area=dst*imarea
c
      A=(rmaj*sin(angle))**2+(rmin*cos(angle))**2
      B=2.D0/dst*(rmin**2-rmaj**2)*sin(angle)*cos(angle)
      C=1.D0/dst**2*((rmaj*cos(angle))**2+(rmin*sin(angle))**2)
```

```fortran
c
      write (*,*) 'A=',a,'    B=',b,'    C=',c
      write (*,*) 'r=',dst

      if (A.eq.C) then
          ang=3.1415926535D0/2.D0
      else
          ang=0.5D0*atan(B/(A-C))
      end if
      write (*,*) 'angle=',ang

c------FIND ACTUAL MAJOR RADII
      rmaja=rmaj*rmin/
     * sqrt(A*(cos(ang))**2+B*sin(ang)*cos(ang)+C*(sin(ang))**2)
      rmina=rmaj*rmin/
     * sqrt(A*(sin(ang))**2-B*sin(ang)*cos(ang)+C*(cos(ang))**2)
c
      if (rmaja.ge.rmina) then
          maj=2.D0*rmaja
      else
          maj=2.D0*rmina
      end if

c      write (7,*) A,B,C,rmaja*2.D0,rmina*2.D0

      RETURN
      END

      SUBROUTINE print_out (vert, area, maj, angle,title)
C--------------------------------------------------------------
C------PRINT STUFF TO SCREEN, FOR DOUBLE CHECKING
c------For a more complete display, remove the comments from
c------the following lines: 249, 314, 383, 384, 399, 400

      implicit real*8 (a-z)
      character title*15
      real*8 vert, area, maj, angle

      write (*,1) title
      write (*,*) '   vert. pos.    area          maj
     *   angle'
      write (*,*) vert,area, maj, angle
 1    format (a)
      RETURN
      END

      SUBROUTINE mark (file,nbub)
C-------------------------------------------------------------
C------SET FILE FOR MARK
      implicit real*8 (a-z)
```

50

```fortran
      integer count,nbub,a
      character film*7, bcolor*6
      common /f/film
      real k,l,m,r,s,t,u,v,lim,od
      logical varblank
      common /vars/ k,l,m,r,s,t,u,v,lim,od,varblank
      character mfile*34, file*34

C------COLOR
      if (file(1:1).eq.'b') bcolor='black'
      if (file(1:1).eq.'g') bcolor='green'
      if (file(1:1).eq.'r') bcolor='red'

C------SCALE
      scale=0.451D0*25.4D0/od

C------OPEN FILES
      mfile='f:\easydij\data\'
      mfile(nblank(mfile)+1:34)=film
      len=(nblank(mfile)+1)
      mfile(len:len)=bcolor(1:1)
      mfile(len+1:34)='.DAT'
      write (*,*) mfile

      open (9, mfile)
      rewind 6


C------TITLE LINES
      write (9,*) film, bcolor, scale, nbub
      write (9,*) '   #          area (sq mm)         maj dia (mm)'
      read (6,50)
50    format (/)

C------TRANSFER DATA
      do 100 count=1,nbub
         read (6,*) a,b,c
         write (9,*) a,b,c
100   continue
      close (9)
      RETURN
      END
```

51

R11#1A   < - film number
6.102e-3
-1.224e-2
0.7216
2.317e-14
5.609
0
8.141e-2
0.452

4.3
25.375   < - outer diameter (in)

APPENDIX A.3 - FORTRAN CODE FOR VISDAT.FOR

```fortran
      PROGRAM VISDAT
C------------------------------------------------------------------
C
C                         VISDAT.FOR
C     DATA REDUCTION PROGRAM FOR VISUALIZATION DATA OBTAINED FROM
C       TRACE
C    OF 16mm HIGH SPEED FILM.  PROGRAM BY MARK A. KEDZIERSKI, APRIL 1991
C
C------------------------------------------------------------------
      IMPLICIT REAL*8(A-Z)
      INTEGER NBUB, FLAGR, NFILES, NBLKFIL, FLAGC,
     *       FLAGS, FLAGD
      DIMENSION AREA(200), MAJDIA(200), VSPHR(200), VELPS(200),
     *        AREAM(200), MAJDIAM(200), QSPHR(200), QELPS(200),
     *        ASUR(200), MINDIAM(200)
      CHARACTER*5 BCOLOR, FILM*7, FIL*12, FILES*12(200), DRIVE*1,
     *         BFIL*12(33), RFIL*12(33), GFIL*12(33), NAME*12
      COMMON /PRT/ FILM,BCOLOR,NBUB,AREA,MAJDIA,VSPHR,VELPS,
     * QSPHR,QELPS,QTSPHR,QTELPS,MINDIAM,ASUR,AREAM,MAJDIAM
      COMMON /FILLOC/ DRIVE
      COMMON /PRTANAL/ NAME, MQVSA, BQVSA, NFRAME, FPS, NUMBLK,
NUMRED,
     *      NSITE, QNUC, FEQSITE, QSUPB, QSUPR, SUPB, SUPR, SUPT,
     *      JPERBW, QGEN, QSUP, EQNUC, RQNUC, EQWALL, QSUP2, EQNUC2,
     *      RQNUC2, HTLGTH, DTSAT, QFLXT, DJDFRM
      COMMON /PRTANL2/ EHCONV, EHNUC, EQFLX, ERRQ, BFLUX, HNUC,
HCONV,
     *      QFLXN, QFLXC, QCORR, AT, QT, DELQ, FLUXT
      COMMON /SCALE/ SCALE
      COMMON /PROP/ DENSV, HFG, TSAT
      WRITE(*,5)
5     FORMAT(' ',5X,'INPUT DRIVE FOR DATA (C,D,E, etc.): ')
      READ(*,*) DRIVE
      WRITE(*,40) DRIVE
40    FORMAT(' ',5X,'REDUCE INDIVIDUAL FILE OR ALL OF THE',/,
     *6X,'FILES IN ',A1,':\EASYDIJ\DATA DIRECTORY (ONE=1 OR ALL=0): ')
      READ(*,20) FLAGR
C
      WRITE(*,300)
300   FORMAT(' ',5X,'READ FPS, ETC. FROM FILE OR INPUT TO SCRN',1X,
     *       '(FILE INPT=1 OR SCRN INPT=0): ')
      READ(*,*) FLAGD
C=====================================================================
====================
C-----REDUCE ONE FILE (FLAGR = 1)
      IF(FLAGR.EQ.1)THEN
20    FORMAT(I2)
      WRITE(*,30)
```

```fortran
30      FORMAT(' ',5X,'INPUT FILE NAME, e.g., R123#4R.DAT: ')
        READ(*,*) FIL
        WRITE(*,31) FIL
31      FORMAT(' ',5X,'Tsat (K) FOR TEST ',A12,': ')
        READ(*,*) TSAT
C
        CALL REDFIL(FIL,FILM,BCOLOR,SCALE,AREA,MAJDIA,NBUB)
        CALL REDDAT
        CALL PRTOUT(6,' ')
        CALL PRTOUT(7,' ')
        CALL PRTOUT(5,FIL)
        OPEN(7,FILE='PRN')
        WRITE(7,200)
        CLOSE(7)
      END IF
C=========================================================
=========================
C
C-----REDUCE ALL THE FILES (FLAGR = 0)
      IF(FLAGR.EQ.0)THEN
        WRITE(*,220)
220     FORMAT(' ',5X,'ANALYZE ALL FILES IN DIR OR CONFIRM EACH'
     *        ,1X,'(ALL=1 OR CONFIRM EACH=0): ')
        READ(*,20) FLAGC
C-------OBTAIN A LIST OF ALL THE DATA FILES IN E:\EASYDIJ\DATA
        CALL DIRFILE(DRIVE//':\EASYDIJ\DATA',FILES,NFILES)
C-------DIVIDE THE FILES UP INTO BLACK, RED AND GREEN BUBBLE FILES
        CALL SORT(FILES,NFILES,BFIL,RFIL,GFIL,NBLKFIL)
        DO 50 I=1, NBLKFIL
          IF(FLAGC.EQ.0)THEN
          WRITE(*,230) BFIL(I)
230       FORMAT(' ',5X,'ANALYZE OR SKIP ',A12,'(1 OR 0): ')
          READ(*,20) FLAGS
          IF(FLAGS.EQ.0)GOTO 240
          END IF
C-------RETRIVE THE DATA
          CALL FLNGTH(BFIL(I),'B',FLGTH)
          NAME = BFIL(I)
          IF(FLAGD.EQ.0)THEN
          CALL FLMDAT(NAME,FLGTH,NFRAME,FPS,MQVSA,BQVSA,NSITE
     *          HTLGTH,DTSAT,QFLXT,DJDFRM,TSAT)
          ELSE
          CALL INFPS(NAME,FLGTH,NFRAME,FPS,MQVSA,BQVSA,NSITE,
     *          HTLGTH,DTSAT,QFLXT,DJDFRM,TSAT)
          END IF
C
          CALL INPUTFIL(BFIL(I),FILM,BCOLOR,SCALE,AREA,MAJDIA,NBUB)
          CALL REDDAT
          CALL PRTOUT(6,' ')
          CALL PRTOUT(7,' ')
```

```
          CALL PRTOUT(5,BFIL(I))
C--------CAPTURE AVERAGE OR TOTAL QUANTIES PERTAINING TO BLACK BUBBLES
          QTBLK = QTELPS
          ASBLK = ASUR(200)
          NUMBLK = NBUB
          VBAVG = VELPS(200)
C         JPERBB = QELPS(200)
C
C

          IF(NAME(1:FLGTH).EQ.'r11#1a'.OR.NAME(1:FLGTH).EQ.'r11#4a')THEN
          QTRED = 0.D0
          ASRED = 0.D0
          NUMRED = 0.D0
          JPERRB = 0.D0
          VRAVG = VBAVG
          ELSE
          CALL INPUTFIL(NAME(1:FLGTH)//'R.DAT',FILM,BCOLOR,SCALE,
     *          AREA,MAJDIA,NBUB)
          CALL REDDAT
          CALL PRTOUT(6,' ')
          CALL PRTOUT(7,' ')
          CALL PRTOUT(5,NAME(1:FLGTH)//'R.DAT')
C-----------CAPTURE AVERAGE OR TOTAL QUANTIES PERTAINING TO RED BUBBLES
          QTRED = QTELPS
          ASRED = ASUR(200)
          NUMRED = NBUB
          JPERRB = QELPS(200)
          VRAVG = VELPS(200)
          END IF
C
C---------CALC THE TOTAL HEAT LOAD TO MAKE ALL THE BUBBLES: QNUC (W)
          QNUC = (QTBLK - QTRED)*FPS/NFRAME
C---------CALC THE BUBBLE FREQ TIMES THE TOTAL # OF SITES: FEQSITE
C---------(BUBBLE/SECOND)
          FEQSITE = (NUMBLK - NUMRED)*FPS/NFRAME
C---------CALC THE AVERAGE HEAT FLUX TO BUBBLE AFTER ITS RELEASE: QSUP
C---------(W/m^2)
          QSUPB = MQVSA*ASBLK + BQVSA
          QSUPR = MQVSA*ASRED + BQVSA
C---------CALC THE SUPERHEAT PER BUBBLE: SUPB (W)
          SUPB = QSUPB*ASBLK
          SUPR = QSUPR*ASRED
C---------CALC THE TOTAL SUPERHEAT FOR THE NET BUBBLES: SUPT (W)
          SUPT = SUPB*(NUMBLK - NUMRED)
C---------SOLVE FOR JOULES PER BUBBLE AT WALL: JPERBW
          JPERBW = (QNUC - SUPT)/FEQSITE
C---------ESTIMATE ENERGY REQUIRED TO NUCLEATE BUB: QGEN (W)
          QGEN = FEQSITE*JPERRB
C---------ESTIMATE ENERGY REQUIRED TO SUPERHEAT AFTER RELEASE: QSUP (W)
          QSUP = SUPT
```

```
C---------ESTIMATE TOTAL NUCLEATIVE ENERGY: EQNUC (W)
          EQNUC = QGEN + QSUP
          RQNUC = EQNUC/QNUC
C---------ESTIMATE THE HEAT FLUX TO THE BUBBLES AT THE WALL, GUESS
C---------THAT 1/2 OF THE BUBBLE SURFACE IS IN CONTACT WITH THE WALL
          IF(ASRED.NE.0.D0)EQWALL = QGEN*2.D0/(ASRED*NUMRED)
C---------ASSUME THAT ALL BLACK BUBBLES CONTRIBUTE TO SUPERHEAT
C---------ESTIMATE ENERGY REQUIRED TO SUPERHEAT AFTER RELEASE: QSUP2
(W)
          QSUP2 = SUPB*NUMBLK
C---------ESTIMATE TOTAL NUCLEATIVE ENERGY: EQNUC (W)
          EQNUC2 = QGEN + QSUP2
          RQNUC2 = EQNUC2/QNUC
C---------CALC THE AMOUNT OF SUBCOOLING OR SUPERHEATING TO BLACK BUBS
          DELQ = -0.5D0*DJDFRM*FPS*(NUMBLK - NUMRED)
C---------CORRECT THE QNUC FOR SUPERHEATING OR SUBCOOLING SO THAT
C---------ONLY THE HEAT REMOVED FROM THE WALL BY NUCLEATION IS AC
C---------ACCOUNTED FOR: QCORR (W)
          QCORR = DELQ + QNUC
C---------CALCULATE THE TOTAL HEAT TO THE ANALYZED AREA: QT (W)
          AT = 0.003048D0*HTLGTH
          QT = QFLXT*AT
C---------CALC THE AMOUNT OF HEAT DISSIPATED BY CONVECTION: QCONV (W)
          QCONV = QT - QCORR
C---------CALC THE CONVECTIVE HEAT FLUX: QFLXC (W/m2)
          QFLXC = QCONV/AT
C---------CALC THE NUCLEATIVE HEAT FLUX: QFLXN (W/m2)
          QFLXN = QCORR/AT
C---------CALC THE CONVECTIVE HEAT TRANSFER COEF: HCONV (W/m2)
          HCONV = QFLXC/DTSAT
C---------CALC THE NUCLEATIVE HEAT TRANSFER COEF: HNUC (W/m2)
          HNUC = QFLXN/DTSAT
C---------CALC THE TOTAL BUBBLE FLUX PER UNIT AREA: BFLUX (BUB/(s m2)
          BFLUX = FEQSITE/AT
C---------CALC THE UNCERTAINTY OF THE QNUC MEAS FOR
C---------95.5% CONFIDENCE INTERVAL: ERRQ (W)
          ERRQ = DSQRT((FPS*0.22D0*QTBLK/NFRAME)**2 + (FPS*0.22D0
     *        *QTRED/NFRAME)**2 + ((QTBLK - QTRED)/NFRAME)**2*80**2)
C---------CALC THE ERROR FOR 95.5% CONF. IN THE HEAT FLUX MEAS: EQFLX
          EQFLX = ERRQ/AT
C---------CALC THE ERROR IN THE HNUC CALC: EHNUC (W/m2 K)
          EHNUC = HNUC*EQFLX/QFLXN
C---------CALC THE ERROR FOR 95.5% CONF. IN THE HCONV CALC: EHCONV
          EHCONV = HCONV*EQFLX/QFLXC
C---------CALC THE BUBFREQ FROM THE QFLXN AND VRAVG
          FLUXT = QFLXN/(VRAVG*DENSV*HFG)
          CALL PRTANAL(6)
          CALL PRTANAL(7)
          CALL PRTANAL(5)
 240      CONTINUE
```

```
 50     CONTINUE
        END IF
200  FORMAT('1')
        STOP
        END
C
C
        SUBROUTINE REDFIL(FIL,FILM,BCOLOR,SCALE,AREA,MAJDIA,NBUB)
C----------------------------------------------------------------------
C
C                  SUBROUTINE REDFIL
C    READS PREEXISTING VISUALIZATION DATA FROM A FILE MADE FROM THE
C    EASYDIJ CAD SYSTEM.
C    THE NAME OF THE FILE IS SET BY THE NAME OF THE REFRIGERANT, THE
C    FILM NUMBER AND THE COLOR OF THE BUBLES. FILE NAME = REF+#FILM
C    NUMBER+FIRST LETTER OF BUBBLE COLOR+.DAT.  PROGRAM BY MARK A.
C    KEDZIERSKI, APRIL 1991.
C
C----------------------------------------------------------------------
        IMPLICIT REAL*8(A-Z)
        INTEGER NBUB, I, FLAGC, NFIX, N
        DIMENSION AREA(200), MAJDIA(200)
        CHARACTER*5 BCOLOR, FILM*7, FIL*12, DRIVE*1, SCRAP*60
        COMMON /FILLOC/ DRIVE
        OPEN(5,FILE=DRIVE//':\EASYDIJ\DATA\'//FIL,STATUS='OLD',ERR=20)
        READ(5,130) FILM, BCOLOR, SCALE, NBUB
130  FORMAT(A7,A5,3X,D22.15,10X,I3)
C-----READ HEADINGS AND PUT IN SCRAP
        READ(5,*) SCRAP
        WRITE(*,*)'READING ',SCRAP
        DO 10 I=1,NBUB
C-------AREA IN mm2 AND MAJDIA IN mm
           READ(5,*) N, AREA(I), MAJDIA(I)
           IF(N.NE.I)WRITE(*,*)'BUBBLE #S DO NOT JIBE'
 10     CONTINUE
        CALL SYSTEM('CLS')
        CLOSE(5)
        WRITE(*,70)
 70     FORMAT(' ',5X,'ANY CORRECTIONS TO THE DATA? (Y->1,N->0): ')
        READ(*,*) FLAGC
        IF(FLAGC.EQ.1)THEN
           WRITE(*,50)
 50        FORMAT(' ',5X,'HOW MANY CORRECTIONS? (INTEGER): ')
           READ(*,*) NFIX
           DO 60 J=1,NFIX
              WRITE(*,40)
 40        FORMAT(' ',5X,'WHICH BUBBLE # TO CORRECT? (INTEGER): ')
              READ(*,*) I
              WRITE(*,90) I
 90        FORMAT(' ',5X,'A(',I2,'): ')
```

```fortran
          READ(*,*) AREA(I)
          WRITE(*,110) I
110       FORMAT(' ',5X,'MAJDIA(',I2,'): ')
          READ(*,*) MAJDIA(I)
60     CONTINUE
          OPEN(5,FILE=DRIVE//':\EASYDIJ\DATA\'//FIL)
          WRITE(5,*) FILM, BCOLOR, SCALE, NBUB
          DO 120 I=1,NBUB
          WRITE(5,*) AREA(I), MAJDIA(I)
120    CONTINUE
          CLOSE(5)
       END IF
       RETURN
20     STOP 'FILE DOES NOT EXIST (MAK IN REDFIL)'
       END
C
C
       SUBROUTINE FLNGTH(FIL,MARK,FLGTH)
C-----------------------------------------------------------------------
C
C                    SUBROUTINE FLNGTH
C    DETERMINES THE LENGTH OF A FILE (FIL) UP TO, BUT NOT INCLUDING
C    THE CHARACTER MARK.
C    OUTPUTS THE LENGTH OF THE FILE THROUGHT THE VARIABLE FLGTH.
C    PROGRAM BY MARK A. KEDZIERSKI, APRIL 1991.
C
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-Z)
       INTEGER COUNT
       CHARACTER*12 FIL, MARK*1
       COUNT=1
20     IF(FIL(COUNT:COUNT).EQ.MARK)GOTO 10
       COUNT = COUNT+1
       GOTO 20
10     FLGTH = COUNT-1
       RETURN
       END
C
C
       SUBROUTINE BUBVOL(NBUB,AREA,MAJDIA,VSPHR,VELPS)
C-----------------------------------------------------------------------
C
C                    SUBROUTINE BUBVOL
C    CALCULATES THE VOLUME OF EACH BUBBLE FOR BOTH SPHERICAL AND
C    ELLIPTICAL SHAPES.  PROGRAM BY MARK A. KEDZIERSKI, APRIL 1991.
C
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-Z)
       INTEGER NBUB, I
       DIMENSION AREA(200), MAJDIA(200), VSPHR(200), VELPS(200)
```

```fortran
      PIE = DACOS(-1.D0)
      DO 10 I=1, NBUB
        VSPHR(I)=4.D0*AREA(I)**1.5/(3.D0*DSQRT(PIE))
        VELPS(I)=8.D0*AREA(I)**2/(3.D0*PIE*MAJDIA(I))
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CONVRT(NBUB,SCALE,AMMSQ,LMM,AMSQ,LM)
C--------------------------------------------------------------------
C
C                  SUBROUTINE CONVRT
C
C   CONVERTS THE INPUT FROM EASYDIJ FROM MM TO METERS.  EASYDIJ HAS
C   ALREADY SCALED AND OPTICALLY CORRECTED THE DATA.  PROGRAM
C   BY MARK A. KEDZIERSKI, APRIL 1991.
C
C--------------------------------------------------------------------
      IMPLICIT REAL*8(A-Z)
      INTEGER NBUB, I
      DIMENSION AMMSQ(200), AMSQ(200), LMM(200), LM(200)
      DO 10 I=1,NBUB
        LM(I) = LMM(I)/1000.D0
        AMSQ(I) =  AMMSQ(I)*1.D-06
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PRTOUT(UNIT,NAME)
C--------------------------------------------------------------------
C
C                  SUBROUTINE PRTOUT
C
C   PRINTS THE INPUT AND REDUCED DATA TO THE SCREEN AND THE PRINTER.
C   PROGRAM BY MARK A. KEDZIERSKI, APRIL 1991.
C
C--------------------------------------------------------------------
      IMPLICIT REAL*8(A-Z)
      INTEGER NBUB, I, UNIT, ERRNUM
      DIMENSION AREA(200), MAJDIA(200), VSPHR(200), VELPS(200),
     *      QSPHR(200), QELPS(200), MINDIAM(200), ASUR(200),
     *      MAJDIAM(200), AREAM(200)
      CHARACTER*5 BCOLOR, FILM*7, TODAY*8, NAME*12, DRIVE*1
      COMMON /PRT/ FILM,BCOLOR,NBUB,AREA,MAJDIA,VSPHR,VELPS,
     * QSPHR,QELPS,QTSPHR,QTELPS,MINDIAM,ASUR,AREAM,MAJDIAM
      COMMON /PROP/ DENSV, HFG, TSAT
      COMMON /FILLOC/ DRIVE
      COMMON /DIA/ DEQ, SIGD, SIGV
```

```fortran
      COMMON /SCALE/ SCALE
      IF(UNIT.EQ.7)OPEN(7,FILE='PRN')
      IF(UNIT.EQ.5)OPEN(5,FILE=DRIVE//':\FORPRG\DATA\'
     * //NAME,ERR=100,IOSTAT=ERRNUM)
      CALL DATE(TODAY)
      WRITE(UNIT,5)TODAY
5     FORMAT('1',64X,A8)
      WRITE(UNIT,10)NBUB, BCOLOR, FILM, SCALE
10    FORMAT(' ',21X,I2,1X,A5,1X,'BUBBLES FROM FILM',1X,A7,/,
     *      23X,'SCALE: 1" = ',E13.7,' mm',//)
      WRITE(UNIT,50)
50    FORMAT(' ',1X,"#",5X,'A',5X,'Majdia',3X,'Mindia',5X,'Asurellp'
     *  ,2X,'Vsp/Vel',2X,'Vellipse',4X,'Qellipse',/,6X,'(mm2)',4X,
     *   '(mm)',6X,'(m)',8X,'(m2)',15X,'(m3)',8X,'(J)',//)
      DO 120 I=1,NBUB
        VRATIO = VSPHR(I)/VELPS(I)
        WRITE(UNIT,40) I, AREA(I), MAJDIA(I), MINDIAM(I), ASUR(I),
     *          VRATIO, VELPS(I), QELPS(I)
40    FORMAT(' ',1X,I2,2X,F5.2,3X,F5.3,3X,E9.3,3X,E9.3,3X,F4.2,3X,
     *      E9.3,3X,E9.3)
120   CONTINUE
      WRITE(UNIT,60) QTSPHR, QTELPS, DENSV, HFG, TSAT
60    FORMAT(' ',///,5X,'HEAT LOAD OF ALL BUBBLES:',//,5X,'ASSUMED',1X
     *,'SPHERICAL (J): ',E15.6,/,5X,'ASSUMED ELLIPTICAL (J): ',E15.6,/
     *,5X,'DENSITY OF VAPOR (Kg/m3): ',F5.3,/,5X,'LATENT HEAT (J/Kg): '
     *,F7.0,/,5X,'SATURATION TEMP. (K): ',F6.2)
      WRITE(UNIT,70) AREAM(200), MAJDIAM(200), MINDIAM(200), ASUR(200)
     *          ,VELPS(200), QELPS(200)
70    FORMAT(' ',///,5X,'AVERAGED QUANTITIES:',//,5X,'CROSS SECTIONAL',
     * 1X,'AREA (m2): ',E15.6,/,5X,'MAJOR DIAMETER (m): ',E15.6,/,5X,
     *'MINOR DIAMETER (m): ',E15.6,/,5X,'SUR. AREA OF ELLIPSOID (m2): '
     *,E15.6,/,5X,'VOLUME OF ELLIPSOID (m3): ',E15.6,/,5X,
     * 'REQUIRED ENERGY TO MAKE ELLIPSOID (J/BUB): ',E15.6)
      WRITE(UNIT,130) DEQ, SIGD, SIGV
130   FORMAT(' ',5X,'EQUIVALENT SPHERICAL DIA FROM ELIPSE VOL (m): ',
     * E15.6,/,5X,'ONE STANDARD DEVIATION OF OF equiv dia (m): ',
     * E15.6,/,5X,'ONE STANDARD DEVIATION OF VOL OF ELIPSE (m3): ',
     * E15.6)
      IF(UNIT.EQ.7)CLOSE(7)
      IF(UNIT.EQ.5)CLOSE(5)
100   IF(ERRNUM.NE.0)WRITE(*,*)'OPEN STATEMENT ERROR #: ',
     *          ERRNUM
      RETURN
      END
C
C
      SUBROUTINE LATENT(FILM,NBUB,VSPHR,VELPS,QSPHR,QELPS,QTSPHR,
     *          QTELPS)
C----------------------------------------------------------------
C
```

```fortran
C                 SUBROUTINE LATENT
C
C         CALCULATES THE AMOUNT OF ENERGY USED TO GENERATE THE
VISUALIZED
C    BUBBLE.  CALCULATIONS ARE DONE ASSUMING BOTH SPHERICAL AND
C    ELLIPTICAL BUBBLE SHAPES.  PROGRAM BY MARK A. KEDZIERSKI, APRIL
C    1991
C
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-Z)
      INTEGER I, NBUB
      CHARACTER*7 FILM
      DIMENSION VSPHR(200), VELPS(200), QSPHR(200), QELPS(200)
      COMMON /PROP/ DENSV, HFG, TSAT
C-----CALC THE DENSITY OF THE VAPOR: DENSV (kg/m3)
C-----CALC THE LATENT HEAT OF VAPORIZATION: HFG (J/Kg)
      CALL FLNGTH(FILM,'#',FLGTH)
      IF(FILM(1:FLGTH).EQ.'R123')THEN
        DENSV=2.6795152D-3*TSAT**2 - 1.379728248D0*TSAT + 179.06845D0
        HFG=-.98076923D0*TSAT**2 + 187.431319D0*TSAT + 202939.835D0
      END IF
      IF(FILM(1:FLGTH).EQ.'R11')THEN
        DENSV=2.5982882D-3*TSAT**2 - 1.334749534D0*TSAT + 173.0516D0
        HFG=-0.651437929D0*TSAT**2 + 3.90200809D0*TSAT + 238129.844D0
      END IF
C-----QTSPHR AND QTELPS IS THE TOTAL ENERGY OF ALL NBUB BUBBLES FOR
C-----SPERICAL AND ELLIPTICAL BUBBLES, RESPECTIVELY
      QTSPHR=0.D0
      QTELPS=0.D0
C-----QSPHR IS THE HEAT REQUIRED FOR THE SPHERICAL BUBBLE I (J)
C-----QELPS IS THE HEAT REQUIRED FOR THE ELLIPTICAL BUBBLE I (J)
      DO 30 I=1, NBUB
        QSPHR(I)=VSPHR(I)*DENSV*HFG
        QELPS(I)=VELPS(I)*DENSV*HFG
        QTSPHR = QTSPHR + QSPHR(I)
        QTELPS = QTELPS + QELPS(I)
 30   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SURAREA(NBUB,AREA,MAJDIA,MINDIA,ASUR)
C-----------------------------------------------------------------------
C
C                 SUBROUTINE SURAREA
C    CALCULATES THE SURFACE AREA OF EACH BUBBLE ASSUMMING AN
C    ELLIPSOIDIAL SHAPE.  PROGRAM BY MARK A. KEDZIERSKI, JULY 1991.
C
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-Z)
```

61

```fortran
      INTEGER NBUB, I
      DIMENSION AREA(200), MAJDIA(200), MINDIA(200), ASUR(200)
      PIE = DACOS(-1.D0)
      DO 10 I=1, NBUB
        MINDIA(I) = 4.D0*AREA(I)/(PIE*MAJDIA(I))
        A = MAJDIA(I)/2.D0
        B = MINDIA(I)/2.D0
        IF(A.LE.B)THEN
C----------ASSUME THAT B AND A ARE VERY CLOSE AND THE BUBBLE IS
C----------SPHERICAL
          ASUR(I) = 4.D0*PIE*B**2
        ELSE
C----------SURFACE IS AN ELLIPSOID: A IS THE MAJOR RADIUS
          R = B/A
          ASUR(I) = 2.D0*PIE*B*A*(R + DACOS(R)/DSQRT(1.D0 - R**2))
        END IF
 10   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE STD(NBUB,X,XAVG,SIG)
      IMPLICIT REAL*8(A-Z)
      INTEGER NBUB
      DIMENSION X(200)
      XSUM = 0.D0
      DO 10 I=1, NBUB
        XSUM = XSUM + X(I)
 10   CONTINUE
      XAVG = XSUM/DFLOAT(NBUB)
      SUM = 0.D0
      DO 20 I=1, NBUB
        SUM = SUM + (XAVG - X(I))**2
 20   CONTINUE
      SIG = 0.D0
      IF(NBUB.GT.1)SIG = DSQRT(SUM/DFLOAT(NBUB - 1))
      RETURN
      END


C
C
      SUBROUTINE AVERAGE(NBUB,AREA,MAJDIA,MINDIA,ASUR,VELPS,QELPS)
      IMPLICIT REAL*8(A-Z)
      INTEGER NBUB, I
      DIMENSION AREA(200), MAJDIA(200), MINDIA(200), ASUR(200),
     *        VELPS(200), QELPS(200)
      ASUM = 0.D0
      MAJSUM = 0.D0
      MINSUM = 0.D0
      SSUM = 0.D0
```

62

```
      VSUM = 0.D0
      QSUM = 0.D0
      DO 10 I=1, NBUB
        ASUM = AREA(I) + ASUM
        MAJSUM = MAJDIA(I) + MAJSUM
        MINSUM = MINDIA(I) + MINSUM
        SSUM = ASUR(I) + SSUM
        VSUM = VELPS(I) + VSUM
        QSUM = QELPS(I) + QSUM
 10   CONTINUE
      AREA(200) = ASUM/NBUB
      MAJDIA(200) = MAJSUM/NBUB
      MINDIA(200) = MINSUM/NBUB
      ASUR(200) = SSUM/NBUB
      VELPS(200) = VSUM/NBUB
      QELPS(200) = QSUM/NBUB
      RETURN
      END
C
C

      SUBROUTINE DIRFILE(DOSDIR,FILES,NFILES)
      IMPLICIT REAL*8(A-Z)
      INTEGER COUNT, NFILES
      CHARACTER *15 DOSDIR, NAME*9, EXT*3, FILES*12(200)
C-----------------------------------------------------------------------
C
C                 SUBROUTINE DIRFILE
C
C   READS THE DOS DIRECTORY AND PUTS THE LIST OF FILES INTO AN ARRAY
C   PROGRAM BY MARK A. KEDZIERSKI, SEPTEMBER 1991.
C
C-----------------------------------------------------------------------
      CALL SYSTEM('DIR '//DOSDIR//' >'//DOSDIR//'\DOSFLST')
      OPEN(5,FILE=DOSDIR//'\DOSFLST',STATUS='OLD',ERR=20)
      COUNT = 0
 50   READ(5,30,ERR=20,END=40) NAME, EXT
      WRITE(*,*) NAME, EXT
      IF(EXT.EQ.'DAT')THEN
        COUNT = COUNT + 1
        CALL FLNGTH(NAME,' ',FLGTH)
        FILES(COUNT)=NAME(1:FLGTH)//'.DAT'
      END IF
      GOTO 50
 40   CONTINUE
      NFILES = COUNT
 30   FORMAT(A9,A3)
      CLOSE(5)
      RETURN
 20   STOP 'FILE DOES NOT EXIST (MAK, SUB DIRFILE)'
      END
```

63

```
C
C
          SUBROUTINE INPUTFIL(FIL,FILM,BCOLOR,SCALE,AREA,MAJDIA,NBUB)
C-------------------------------------------------------------------
C
C                      SUBROUTINE INPUTFIL
C     READS PREEXISTING VISUALIZATION DATA FROM A FILE MADE WITH THE
C     EASYDIJ SUBROUTINE OF THE VISDAT.FOR PROGRAM.  PROGRAM BY MARK
A.
C     KEDZIERSKI, SEPT. 1991.
C
C-------------------------------------------------------------------
          IMPLICIT REAL*8(A-Z)
          INTEGER NBUB, I, N
          DIMENSION AREA(200), MAJDIA(200)
          CHARACTER*5 BCOLOR, FILM*7, FIL*12, DRIVE*1, SCRAP*60
          COMMON /FILLOC/ DRIVE
          CALL FLNGTH(FIL,'.',FLGTH)
          OPEN(5,FILE=DRIVE//':\EASYDIJ\DATA\'//FIL(1:FLGTH+4),STATUS='OLD'
     *     ,ERR=20)
          READ(5,40) FILM, BCOLOR, SCALE, NBUB
          WRITE(*,*) NBUB, BCOLOR
C-----READ THE HEADINGS AND PUT IN SCRAP
          READ(5,*) SCRAP
          WRITE(*,*)'READING ',SCRAP
          IF(NBUB.EQ.0)GOTO 30
          DO 10 I=1,NBUB
            READ(5,*) N, AREA(I), MAJDIA(I)
            IF(N.NE.I)WRITE(*,*)'BUBBLE #S DO NOT JIBE'
 10     CONTINUE
C      CALL SYSTEM('CLS')
 30     CONTINUE
 40     FORMAT(A7,A5,3X,D22.15,10X,I3)
          CLOSE(5)
          RETURN
 20     STOP 'FILE DOES NOT EXIST (MAK, SUB INPUTFIL)'
          END
C
C
          SUBROUTINE REDDAT
          IMPLICIT REAL*8(A-Z)
          INTEGER NBUB
          DIMENSION AREA(200), MAJDIA(200), VSPHR(200), VELPS(200),
     *         AREAM(200), MAJDIAM(200), QSPHR(200), QELPS(200),
     *         ASUR(200), MINDIAM(200)
          CHARACTER*5 BCOLOR, FILM*7
          COMMON /PRT/ FILM,BCOLOR,NBUB,AREA,MAJDIA,VSPHR,VELPS,
     * QSPHR,QELPS,QTSPHR,QTELPS,MINDIAM,ASUR,AREAM,MAJDIAM
          COMMON /DIA/ DEQ, SIGD, SIGV
          COMMON /SCALE/ SCALE
```

```fortran
      CALL CONVRT(NBUB,SCALE,AREA,MAJDIA,AREAM,MAJDIAM)
      CALL SURAREA(NBUB,AREAM,MAJDIAM,MINDIAM,ASUR)
      CALL BUBVOL(NBUB,AREAM,MAJDIAM,VSPHR,VELPS)
      CALL STD(NBUB,VELPS,VAVG,SIGV)
      CALL LATENT(FILM,NBUB,VSPHR,VELPS,QSPHR,QELPS,QTSPHR,QTELPS)
      CALL AVERAGE(NBUB,AREAM,MAJDIAM,MINDIAM,ASUR,VELPS,QELPS)
C-----CALC AN EQUIVALENT SPHERICAL DIAMETER FROM VELPS: DEQ (m)
      DEQ = (6.D0*VAVG/3.14159D0)**(1./3.)
      SIGD = SIGV*DEQ/(3.D0*VAVG)
      RETURN
      END
C
C
      SUBROUTINE SORT(FILES,NFILES,BFIL,RFIL,GFIL,BCOUNT)
      IMPLICIT REAL*8(A-Z)
      INTEGER I, NFILES, BCOUNT, RCOUNT, GCOUNT
      CHARACTER FILES*12(200), BFIL*12(33), RFIL*12(33), GFIL*12(33),
     *      NAME*12
      BCOUNT = 0
      RCOUNT = 0
      GCOUNT = 0
      DO 10 I=1, NFILES
        CALL FLNGTH(FILES(I),'.',FLGTH)
        NAME = FILES(I)
        IF(NAME(FLGTH:FLGTH).EQ.'B')THEN
          BCOUNT = BCOUNT + 1
          BFIL(BCOUNT) = NAME
        END IF
        IF(NAME(FLGTH:FLGTH).EQ.'R')THEN
          RCOUNT = RCOUNT + 1
          RFIL(RCOUNT) = NAME
        END IF
        IF(NAME(FLGTH:FLGTH).EQ.'G')THEN
          GCOUNT = GCOUNT + 1
          GFIL(GCOUNT) = NAME
        END IF
 10   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FLMDAT(NAME,FLGTH,NFRAME,FPS,MQVSA,BQVSA,NSITE,
     *         HTLGTH,DTSAT,QFLXT,DJDFRM,TSAT)
      IMPLICIT REAL*8(A-Z)
      INTEGER ERRNUM
      CHARACTER NAME*12, DRIVE*1
      COMMON /FILLOC/ DRIVE
      COMMON /SCALE/ SCALE
        CALL SYSTEM('CLS')
        WRITE(*,60) NAME(1:FLGTH)
```

65

```fortran
60      FORMAT(' ',5X,'INPUT DATA FOR FILE: ', A12,/,
     *       6X,'# OF FRAMES BLK BUBBLE PAST RED: ')
        READ(*,*) NFRAME
        WRITE(*,80)
80      FORMAT(' ',5X,'INPUT # OF FRAMES PER SECOND: ')
        READ(*,*) FPS
        WRITE(*,110)
110     FORMAT(' ',5X,'SLOPE OF q VS. Asur FIT (W/m^4): ')
        READ(*,*) MQVSA
        WRITE(*,120)
120     FORMAT(' ',5X,'INTERCEPT OF q VS. Asur FIT (W/m^2): ')
        READ(*,*) BQVSA
        WRITE(*,50)
50      FORMAT(' ',5X,'AVERAGE # OF ACTIVE SITES: ')
        READ(*,*) NSITE
        WRITE(*,130)
130     FORMAT(' ',5X,'LENGTH OF HEATER FROM TRACE (inches): ')
        READ(*,*) HTLGTH
C-------CONVERT THE LENGTH OF THE HEAT TO METERS
        HTLGTH = HTLGTH*SCALE/1000.D0
        WRITE(*,140)
140     FORMAT(' ',5X,'Twall - Tsat (K):  ')
        READ(*,*) DTSAT
        WRITE(*,150)
150     FORMAT(' ',5X,'HEAT FLUX TO TUBE (W/m2): ')
        READ(*,*) QFLXT
        WRITE(*,160)
160     FORMAT(' ',5X,'AVG SLOPE FROM Qbub vs. frames (J/frame): ')
        READ(*,*) DJDFRM
        WRITE(*,40) NAME(1:FLGTH)
40      FORMAT(' ',5X,'Tsat (K) FOR TEST ',A12,': ')
        READ(*,*) TSAT
C-------WRITE THE DATA TO A FILE
        OPEN(5,FILE=DRIVE//':\FORPRG\DATA\COMBINE\'//NAME(1:FLGTH)//
     *  '.FPS',ERR=10,IOSTAT=ERRNUM)
        WRITE(5,*) NAME(1:FLGTH), NFRAME, FPS
        WRITE(5,*) MQVSA, BQVSA, NSITE
        WRITE(5,*) HTLGTH, DTSAT, QFLXT
        WRITE(5,*) DJDFRM, TSAT
        CLOSE(5)
        RETURN
10      WRITE(*,*) 'OPEN STATEMENT ERROR #:',ERRNUM,' IN FLMDAT'
        STOP
        END
C
C
        SUBROUTINE INFPS(NAME,FLGTH,NFRAME,FPS,MQVSA,BQVSA,NSITE,
     *          HTLGTH,DTSAT,QFLXT,DJDFRM,TSAT)
        IMPLICIT REAL*8(A-Z)
        CHARACTER*12 NAME, DRIVE*1
```

```fortran
      COMMON /FILLOC/ DRIVE
      OPEN(5,FILE=DRIVE//':\FORPRG\DATA\COMBINE\'//NAME(1:FLGTH)//
     * '.FPS',STATUS='OLD',ERR=10)
      READ(5,*) NAME(1:FLGTH), NFRAME, FPS
      READ(5,*) MQVSA, BQVSA, NSITE
C-------HTLGTH IN METERS, DTSAT IN K, QFLXT IN W/m2
      READ(5,*) HTLGTH, DTSAT, QFLXT
      READ(5,*) DJDFRM, TSAT
      CLOSE(5)
      RETURN
 10   STOP 'FILE NOT THERE (MAK SUB INFPS)'
      END
C
C
      SUBROUTINE PRTANAL(UNIT)
      IMPLICIT REAL*8 (A-Z)
      INTEGER UNIT, ERRNUM
      CHARACTER NAME*12, DRIVE*1, TODAY*8
      COMMON /PRTANAL/ NAME, MQVSA, BQVSA, NFRAME, FPS, NUMBLK,
     NUMRED,
     *      NSITE, QNUC, FEQSITE, QSUPB, QSUPR, SUPB, SUPR, SUPT,
     *      JPERBW, QGEN, QSUP, EQNUC, RQNUC, EQWALL, QSUP2, EQNUC2,
     *      RQNUC2, HTLGTH, DTSAT, QFLXT, DJDFRM
      COMMON /PRTANL2/ EHCONV, EHNUC, EQFLX, ERRQ, BFLUX, HNUC,
     HCONV,
     *      QFLXN, QFLXC, QCORR, AT, QT, DELQ, FLUXT
      COMMON /FILLOC/ DRIVE
      IF(UNIT.EQ.7)OPEN(7,FILE='PRN')
      IF(UNIT.EQ.5)OPEN(5,FILE=DRIVE//':\FORPRG\DATA\COMBINE\'
     * //NAME,ERR=100,IOSTAT=ERRNUM)
      CALL DATE(TODAY)
      WRITE(UNIT,5)TODAY
 5    FORMAT('1',64X,A8)
      WRITE(UNIT,140) NAME
 140  FORMAT(' ',5X,'ANALYSIS OF ',A8,' DATA',//)
      WRITE(UNIT,170)
 170  FORMAT(' ',20X,'****INPUT DATA****',/)
      WRITE(UNIT,130) MQVSA, BQVSA, NFRAME, FPS
      WRITE(UNIT,190) NUMBLK, NUMRED, NSITE
 190  FORMAT(' ',4X,'NUMBER OF BLACK BUBBLES: ',F4.0,/,
     *          5X,'NUMBER OF RED BUBBLES: ',F4.0,/,
     *          5X,'NUMBER OF ACTIVE SITES: ',F4.0)
      WRITE(UNIT,270) HTLGTH, DTSAT, QFLXT, DJDFRM
 270  FORMAT(' ',4X,'HEATER DATA LENGTH (m): ',F8.2,/,
     *          5X,'Twall - Tsat (K): ',F5.2,/,
     *          5X,'HEAT FLUX FROM HEATER (W/m2): ',F6.0,/,
     *          5X,'AVG dQbub/dframe (J/frame): ',E16.9,//)
      WRITE(UNIT,180)
 180  FORMAT(' ',15X,'****CALCULATED QUANTITIES****',/)
      WRITE(UNIT,90) QNUC, FEQSITE, QSUPB, QSUPR
```

```
 90  FORMAT(' ',/,5X,'HEAT REQ. TO GEN. VAPOR (W): ',F5.3,/,
    * 5X,'BUBBLE FREQ TIMES THE TOTAL # OF SITES (BUB/SEC): ',F6.1,/,
    * 5X,'AVG HEAT FLUX TO BLK BUBBLE AFTER RELEASE (W/m2): ',F7.0,/,
    * 5X,'AVG HEAT FLUX TO RED BUBBLE AFTER RELEASE (W/m2): ',F7.0)
     WRITE(UNIT,150) SUPB, SUPR, SUPT, JPERBW
150  FORMAT(' ',/,5X,'AVG SUPERHEAT TO BLK BUB (W/BUB): ',F5.3,/,
    * 5X,'AVG SUPERHEAT TO RED BUB (W/BUB): ',F5.3,/,
    * 5X,'TOTAL SUPERHEAT TO THE NET BUBBLES (W): ',F5.3,/,
    * 5X,'JOULES PER BUBBLE AT THE WALL (J/BUB): ',F7.5,/)
130  FORMAT(' ',4X,'SLOPE OF q VS. Asur FIT (W/m^4): ',E16.9,/,
    * 5X,'INTERCEPT OF q VS. Asur FIT (W/m^2): ',E16.9,/,
    * 5X,'# OF FRAMES THAT BLK BUBS AFTER RED (-): ',F4.0,/,
    * 5X,'FRAMES PER SECOND: ',F5.0)
     WRITE(UNIT,160) QGEN, QSUP, EQNUC, RQNUC
160  FORMAT(' ',4X,'EST HEAT REQ TO GEN. BUB AT WALL (W): ',F5.3,/,
    * 5X,'ESTIMATED HEAT REQUIRED TO SUPERHEAT BUBS (W): ',F5.3,/,
    * 5X,'ESTIMATED QNUC OR SUM OF PREV. TWO (W): ',F5.3,/,
    * 5X,'RATIO OF EST. QNUC TO QNUC (-): ',F5.3)
     WRITE(UNIT,210) EQWALL
210  FORMAT(' ',4X,'ESTIMATED WALL HEAT FLUX TO RED BUBBLES',/,
    * 5X,'ASS. 1/2 OF Asur OF RED BUB IN CONTACT WITH WALL (W/m2): ',
    * F6.0)
     WRITE(UNIT,260) QSUP2, EQNUC2, RQNUC2
260  FORMAT(' ',//,10X,'****ASS. ALL BLK BUB CONTRIBUTE TO SUP****',
    * /,5X,'ESTIMATED HEAT REQUIRED TO SUPERHEAT BUBS (W): ',F5.3,/,
    * 5X,'ESTIMATED QNUC OR SUM OF PREV. TWO (W): ',F5.3,/,
    * 5X,'RATIO OF EST. QNUC TO QNUC (-): ',F5.3)
     WRITE(UNIT,280) HCONV, EHCONV, HNUC, EHNUC
280  FORMAT(' ',//,10X,'****HEAT TRANSFER CALCS****',/,
    * 5X,'CONVECTIVE HEAT TRANSFER COEF (W/m2K): ',F7.2,/,
    * 5X,'UNCERTAINTY IN Hconv FOR 95.5% CONF (W/m2K): ',F6.2,/,
    * 5X,'NUCLEATIVE HEAT TRANSFER COEF (W/m2K): ',F7.2,/,
    * 5X,'UNCERTAINTY IN hnuc FOR 95.5% CONF (W/m2K): ',F6.2)
     WRITE(UNIT,290) BFLUX, QFLXN, EQFLX, ERRQ
290  FORMAT(' ',4X,'TOTAL # OF BUBBLES PER SEC m2: ',E16.9,/,
    *  5X,'NUCLEATIVE HEAT FLUX (W/m2): ',F8.2,/,
    *  5X,'UNCERTAINTY IN qnuc FOR 95.5% CONF (W/m2): ',F7.2,/,
    *  5X,'UNCERTAINTY IN Qnuc MEAS. FOR 95.5% CONF (W): ',F5.3)
     WRITE(UNIT,300) QFLXC, QCORR, AT, QT, FLUXT
300  FORMAT(' ',4X,'CONVECTIVE HEAT FLUX (W/m2): ',F8.2,/,
    *  5X,'CORRECTED Qnuc: HEAT ONLY TO MAKE BUBS (W): ',F5.3,/,
    *  5X,'ANALYZED HEAT TRANSFER SURFACE AREA (m2): ',E16.9,/,
    *  5X,'TOTAL HEAT INCIDENT ANALYZED AREA (W): ',F5.3,/,
    *  5X,'BUBBLE FLUX: qnuc/(pvhfgVravg) (bub/(sm2): ',E16.9)
     IF(UNIT.EQ.5.OR.UNIT.EQ.7)CLOSE(UNIT)
100  IF(ERRNUM.NE.0)WRITE(*,*)'OPEN STATEMENT ERROR #: ',
    *           ERRNUM
     RETURN
     END
```

APPENDIX B.1 - FORTRAN CODE FOR THE IMAGE PROCESSING PROGRAM

```fortran
C            PROGRAM: IMAGE PROCESSING
C       THIS PROGRAM IS CAPABLE OF SAVING AN IMAGE TO A FILE,
C       RESTORING THE IMAGE FROM A DATA FILE, DETECTING BUBBLE
C       EDGES, CONVERTING AN IMAGE TO A PIXEL DATA FILE AND A
C       PIXEL DATA FILE TO AN IMAGE.
C
        INTEGER*2 ISTAT,XBOXLEN,YBOXLEN,XPOS,YPOS,IVA
        INTEGER*2 PARRAY(20)
        CHARACTER*20 FNAME
C
        OPEN (UNIT=10,FILE='TOP.DAT')
        OPEN (UNIT=11,FILE='BOTTOM.DAT')
        OPEN (UNIT=12,FILE='LEFT.DAT')
        OPEN (UNIT=13,FILE='RIGHT.DAT')
C
C       INITIALIZE, RESET AND DECLARE INPUT/OUTPUT BUFFERS
        CALL ISINIT
        CALL ISRSET
        ISTAT = ISINFR (1)
        ISTAT = ISOTFR (1)
C       CALL TO MENU DRIVER
100     CALL MENU (IRESP)
C
C       ACQUIRE THE IMAGE
C       FIRST SET SYNC EXTERNAL
C       SECOND PUT INTO PASSTHRU
C       THIRD ACQUIRE AN IMAGE
        IF (IRESP.EQ.1) THEN
         ISTAT = ISSYNC (1)
         ISTAT = ISPASS ()
         WRITE (*,10)
10       FORMAT (5X,'THE BOARD IS IN PASSTHRU, PRESS <RETURN> TO
     &  ACQUIRE AN IMAGE.',$)
         READ (*,15) ians
15       FORMAT (a1)
         ISTAT = ISACQ (1,255)
         GOTO 100
        ENDIF
C
C       SAVE THE IMAGE TO A FILE
        IF (IRESP.EQ.2) THEN
          WRITE (*,20)
20       FORMAT (5X,'SAVE THE IMAGE IN A FILE AS: ',$)
          READ (*,30) FNAME
30       FORMAT (A20)
          ISTAT = ISSAVE (1,0,1,0,FNAME)
          ISTAT = ISDISP (1)
          GOTO 100
```

69

```fortran
      ENDIF
C
C     DRAW A BOX
      IF (IRESP.EQ.3) THEN
35      CALL CLS
        WRITE (*,37)
37      FORMAT (5X,'HORIZONTAL LENGTH IS: ',$)
        READ (*,40) XBOXLEN
40      FORMAT (I3)
        WRITE (*,42)
42      FORMAT (5X,'  VERTICAL LENGTH IS: ',$)
        READ (*,40) YBOXLEN
        WRITE (*,44)
44      FORMAT (5X,'ENTER THE STARTING X-POSITION: ',$)
        READ (*,40) XPOS
        WRITE (*,45)
45      FORMAT (5X,'ENTER THE STARTING Y-POSITION: ',$)
        READ (*,40) YPOS
        IF(((XBOXLEN+XPOS).GT.511).OR.((YBOXLEN+YPOS).GT.511))
     &    THEN
          WRITE (*,47)
47        FORMAT (1X,'POSITION TOO LARGE, PRESS <RETURN>',$)
          READ (*,48) IANS
48        FORMAT (A1)
          GOTO 35
        ENDIF
        ISTAT = ISGPOS (YPOS,XPOS)
        DO 49 I = 1, 8
          PARRAY(I*2-1) = YPOS
          PARRAY(I*2) = XPOS
49      CONTINUE
        PARRAY(3) = PARRAY(3) + YBOXLEN
        PARRAY(5) = PARRAY(3)
        PARRAY(6) = PARRAY(6) + XBOXLEN
        PARRAY(8) = PARRAY(6)
        ISTAT = ISLINE (1,8,PARRAY)
        ISTAT = ISDISP (1)
        GOTO 100
      ENDIF
C
C     DETECT THE BUBBLE EDGE POINTS
      IF (IRESP.EQ.4) THEN
        IS = YPOS + 1
        JS = XPOS + 1
        IE = YPOS + YBOXLEN - 1
        JE = XPOS + XBOXLEN - 1
C
        IS1 = IS + 1
        NUM1 = 0
        NUM2 = 0
```

```fortran
          DO 88 J = JS, JE
            II = IE - 1
            ICOUNT = 0
            DO 85 I = IS, II
              ISTAT = ISGETP(1,I,J,1,IVA)
              K = 512 - I
              IF (IVA.EQ.255) THEN
                ICOUNT = ICOUNT + 1
                IF(ICOUNT.EQ.1) THEN
                NUM1 = NUM1 + 1
                WRITE (10,84) J, K
84              FORMAT (5X, 2I5, 5X, 2I7)
                ENDIF
              ENDIF
85          CONTINUE
            ICOUNT = 0
            DO 87 I = II, IS, -1
              ISTAT = ISGETP(1,I,J,1,IVA)
              K = 512 - I
              IF(IVA.EQ.255) THEN
                ICOUNT = ICOUNT + 1
                IF(ICOUNT.EQ.1) THEN
                  NUM2 = NUM2 + 1
                  WRITE(11,84) J, K
                ENDIF
              ENDIF
87          CONTINUE
88        CONTINUE
          WRITE (10,*) NUM1
          WRITE (11,*) NUM2
          GOTO 100
        ENDIF
C
C     RESTORING AN IMAGE FROM DISK
      IF (IRESP.EQ.5) THE
        WRITE (*,90)
90      FORMAT (5X,'ENTER FILE NAME TO BE DISPLAYED: ',$)
        READ (*,30) fname
        ISTAT = ISREST (1,0,0,FNAME)
        ISTAT = ISDISP (1)
        GOTO 100
      ENDIF
C
C     CONVERT A PIXEL DATA FILE TO AN IMAGE
      IF (IRESP.EQ.6) THEN
        READ(12,*) NUM1
        DO 91 K = 1, NUM1
          READ(12,*) X, Y
          I = X
          J = Y
```

```
                J1 = 512 - J
                ISTAT = ISPUTP(1,J1,I,1,255)
91        CONTINUE
          READ(13,*) NUM2
         DO 92 K = 1, NUM2
           READ(13,*) X, Y
           I = X
           J = Y
           J1 = 512 - J
           ISTAT = ISPUTP(1,J1,I,1,255)
92      CONTINUE
        WRITE(*,93)
93      FORMAT(30X,'ENTER FILE NAME TO SAVE AND DISPLAY:'$)
         READ(*,94) FNAME
94      FORMAT(A20)
          ISTAT = ISSAVE(1,0,1,4,FNAME)
          ISTAT = ISDISP(1)
        ENDIF
C
      CALL ISEND
C
      STOP
      END
C****************************************************************
C
      SUBROUTINE MENU (IRESP)
C
C****************************************************************
C
100   CALL CLS
      WRITE (*, 5)
      WRITE (*,10)
      WRITE (*,20)
      WRITE (*,30)
      WRITE (*,40)
      WRITE (*,50)
      WRITE (*,60)
      WRITE (*,70)
      WRITE (*,80)
      READ (*,90) IRESP
C
      IF ((IRESP.LT.1).OR.(IRESP.GT.7)) GOTO 100
       CALL CLS
       WRITE (*,95)
C
5       FORMAT ((/),25X,'DT-IRIS FUNCTIONALITY TEST MENU',(/))
10      FORMAT (30X,' (1) ACQUIRE AN IMAGE',/)
20      FORMAT (30X,' (2) SAVE AN IMAGE TO A DATA FILE',/)
30      FORMAT (30X,' (3) DRAW A BOX',/)
40      FORMAT (30X,' (4) DETECT THE EDGE POINTS',/)
```

```fortran
50      FORMAT (30X,' (5) RESTORE AN IMAGE FROM DISK',/)
60      FORMAT (30X,' (6) CONVERT DATA FILE TO IMAGE',/)
70      FORMAT (30X,' (7) EXIT THIS PROGRAM',/)
80      FORMAT (35X,' MAKE SELECTION: ',$)
90      FORMAT (I1)
95      FORMAT ((/))
        RETURN
        END
C
C*************************************************************
C
        SUBROUTINE CLS
C
C*************************************************************
        character erase(4)
C
        Erase(1) = char(27)
        Erase(2) = char(91)
        Erase(3) = char(90)
        Erase(4) = char(74)
c
        write (*,10) (Erase(i),i=1,4)
10      format (/,1x,4a1,$)
c
        Return
        End
```

73

APPENDIX B.2 - HOW TO USE THE IMAGE PROCESSING PROGRAM

This program has seven functions, acquiring an image, saving an image to a data file, drawing a box, detecting the bubble edge, restoring an image from disk, converting a data file to an image, and exiting the program.

The program is PC-based because it is executed by linking it with DT-IRIS subroutine libraries ISFORLIB.LIB. The program will ask for your selection of the above functions. If you want to save an image to a data file or restore an image from disk, the program will ask for the file name to be saved or restored after your selection of this function. The file is in the format of .img. If you want to draw a box, the program will ask you the dimensions of the rectangular box after your selection of this function. After detecting the bubble edge, the program will save the edge point coordinates into two data files TOP.DAT and BOTTOM.DAT. TOP.DAT is the data file for top edge coordinates and BOTTOM.DAT is the data file for bottom edge coordinates. To convert a data file to an image, you need to enter the data file name. When converting data from multiple-bubble completion, the input data files are LEFT.DAT and RIGHT.DAT. LEFT.DAT is the data file for extension from left and RIGHT.DAT is the data file for extension from right.

Most functions run very quickly (within one minute). Recording the edge point coordinates, however, may take 5 to 30 minutes depending on the size of the image. A typical .img file's size is about 0.26 MB, and the pixel data file size may vary depending on the size of the image. To run this program, a PC with Data Translation board, DT-IRIS subroutine libraries, and a FORTRAN compiler are needed.

APPENDIX C.1 - MULTIPLE-BUBBLE SEPARATION AND COMPLETION PROGRAM

```
      PROGRAM CUSP
C     ---------------------------------------------------------
C     THIS PROGRAM READS IN THE PIXEL DATA FILE GENERATED
C     BY IMAGE PROCESSING PROGRAM, USES LEAST-SQUARE
C     CURVE-C FITTING TO FIND DERIVATIVES TO LOCATE THE
C     CUSPS, THEN EXTENDS TO COMPLE THE BUBBLES.
C     ---------------------------------------------------------
C     PARAMETERS:
C     X, Y - ARRAY OF X AND Y VALUES.
C     X1, Y1 - ARRAY OF X AND Y VALUES FOR EACH CURVE FITTING.
C     MS, MF - THE RANG DEGREE OF POLYNOMIALS.
C             (THE MAXIMUM DEGREE IS 9.)
C
      DOUBLE PRECISION X(500), Y(500)
      INTEGER MS, MF, NUM
C
      OPEN (UNIT=10,FILE='TOP.DAT')
      OPEN (UNIT=11,FILE='BOTTOM.DAT')
      OPEN (UNIT=12,FILE='DERIV.DAT')
      OPEN (UNIT=13,FILE='LEFT.DAT')
      OPEN (UNIT=14,FILE='RIGHT.DAT')
C
C     SPECIFY THE DEGREE OF POLYNOMIALS
      MS = 2
      MF = 2
C
      WRITE(12,10)
10    FORMAT(15X,'X',15X,'Y',15X,'1ST DERIV',15X,'SIGN')
C
C     INPUT DATA FOR THE TOP CURVE
      DO 25 INP  = 10, 11
        READ(INP,*) NUM
        DO 20 I = 1, NUM
          READ(INP,*) KI, KJ
          X(I) = FLOAT(KI)
          Y(I) = FLOAT(KJ)
20      CONTINUE
        CALL LCUSP(X,Y,NUM,MS,MF,INP)
25    CONTINUE
      STOP
      END
C
      SUBROUTINE LCUSP(X,Y,NUM,MS,MF,INP)
C-------------------------------------------------------------
C     THIS SUBROUTINE LOCATES CUSPS.
C-------------------------------------------------------------
      DOUBLE PRECISION X(500),Y(500)
      DOUBLE PRECISION X1(100),Y1(100),S1(500),S2(500)
```

```fortran
      DOUBLE PRECISION C(10)
      DOUBLE PRECISION FY1(500), FY2(500)
      DOUBLE PRECISION aa
C
      INTEGER MS,MF,IM15,IP15,IM30,IP30,NUM,NUMM15
C
      NUMM15 = NUM - 15
      MEP1 = MF + 1
      DO 100 I = 1, NUM
        IM15 = I - 15
        IP15 = I + 15
        IM30 = I - 30
        IP30 = I + 30
        IF (I.LT.16) THEN
          DO 20 J = I, IP30
            X1(J-I+1) = X(J)
            Y1(J-I+1) = Y(J)
20        CONTINUE
          GOTO 43
        ENDIF
        IF (I.GT.NUMM15) THEN
          DO 30 J = IM30, I
            X1(J-IM30+1) = X(J)
            Y1(J-IM30+1) = Y(J)
30        CONTINUE
          GOTO 43
        ENDIF
        DO 40 J = IM15, IP15
          X1(J-IM15+1) = X(J)
          Y1(J-IM15+1) = Y(J)
40      CONTINUE
43      CALL LEASTSQ(X1,Y1,31,C,MS,MF)
C       CALCULATE FIRST ORDER DERIVATIVES
        aa=1.
        FY1(I) = C(2) + 2.0*C(3)*X(I) + 3.0*C(4)*(X(I))**2
        FY2(I) = 2.0*C(3) + 6.0*C(4)*X(I)
        S1(I) = SIGN(aa,FY1(I))
        S2(I) = SIGN(aa,FY2(I))
        WRITE(12,50) I, X(I), Y(I), FY1(I), FY2(I), S1(I)
50      FORMAT(2X, I4, 5F13.3)
100   CONTINUE
C
C       LOCATE THE CUSP
      DO 120 I = 2, NUM
        IF(S1(I-1).NE.S1(I)) THEN
          WRITE(12,110) X(I), Y(I)
110       FORMAT(/5X, 'THE CUSP LOCATION IS:  (', 2F10.2, ')')
          CALL EXTEND(X,Y,I,FY1,FY2,INP)
        ENDIF
120   CONTINUE
```

```fortran
C
      RETURN
      END
C

      SUBROUTINE EXTEND(X,Y,I,FY1,FY2,INP)
C      ----------------------------------------------------
C      EXTEND THE CURVES
C      ----------------------------------------------------
C

      DOUBLE PRECISION X(500), Y(500), FY1(500), FY2(500)
      DOUBLE PRECISION XEL(150), YEL(150), XER(150), YER(150)
      DOUBLE PRECISION DL1(150), DR1(150)
      DOUBLE PRECISION D2
C

      XEL(1) = X(I-1)
      YEL(1) = Y(I-1)
      DL1(1) = FY1(I-10)
      XER(1) = X(I)
      YER(1) = Y(I)
      DR1(1) = FY1(I+10)
      SUM = 0.0
      DO 5 JJ = 1, 15
        SUM = SUM + FY2(I-5-JJ)
5     CONTINUE
      D2 = ABS(SUM/15.0)
      IF (INP.EQ.10) THEN
        D2 = - D2
      ENDIF
      print*, D2
      DO 20 K = 2, 50
C      LEFT POINT EXTENSION
        XEL(K) = XEL(K-1) + 1.0
        YEL(K) = YEL(K-1) + DL1(K-1) + D2/2.0
        DL1(K) = (YEL(K) - YEL(K-1))/1.0
      WRITE(13,10) XEL(K), YEL(K)
C      RIGHT POINT EXTENSION
        XER(K) = XER(K-1) - 1.0
        YER(K) = YER(K-1) - DR1(K-1) + D2/2.0
        DR1(K) = (YER(K-1) - YER(K))/1.0
      WRITE(14,10) XER(K), YER(K)
10      FORMAT(5X, 2F10.2)
20    CONTINUE
C
      RETURN
      END
C
      SUBROUTINE LEASTSQ(X1,Y1,N,C,MS,MF)
C      ---------------------------------------------------------------
C      THIS SUBROUTINE USES LEAST-SQUARE METHOD TO FIT A
C      POLYNOMIAL CURVE.
```

```fortran
C       ---------------------------------------------------------
C       PARAMETERS:
C       X1, Y1 - ARRAY OF X AND Y VALUES FOR EACH CURVE FITTING.
C       N - NUMBER OF DATA PAIRS FOR EACH CURVE FITTING.
C       A - AUGMENTED ARRAY OF THE COEFFICIENTS OF THE NORMAL
C           EQUATIONS
C       C - ARRAY OF COEFFICIENTS OF LEAST SQUARE POLYNOMIALS.
C       ---------------------------------------------------------
        DOUBLE PRECISION X1(500), Y1(500), C(10), A(10,11), XN(500),
     &              SUM, BETA
        INTEGER N, MS, MF, MEP1, MEP2, I, J, IM1, IPT, ICOEF, JCOEF
C
        IF(MF.GT.(N-1)) THEN
          MF = N - 1
          PRINT 200, MF
        ENDIF
5       MEP1 = MF + 1
        MEP2 = MF + 2
        CALL RESET(A,MEP1,MEP2)
C       ---------------------------------------------------------
C       PUT ONES INTO A NEW ARRAY. THIS WILL HOLD THE POWERS
C       OF THE X VALUES AS WE PROCEED.
C       ---------------------------------------------------------
        DO 10 I = 1, N
          XN(I) = 1.0
10      CONTINUE
C       ---------------------------------------------------------
C       COMPUTE FIRST COLUMN AND N+1 ST COLUMN OF A. I MOVES
C       DOWN THE ROWS, J SUMS OVER THE N VALUES.
C       ---------------------------------------------------------
        DO 30 I = 1, MEP1
          A(I,1) = 0.0
          A(I,MEP2) = 0.0
          DO 20 J = 1, N
            A(I,1) = A(I,1) + XN(J)
            A(I,MEP2) = A(I,MEP2) + Y1(J)*XN(J)
            XN(J) = XN(J)*X1(J)
20        CONTINUE
30      CONTINUE
C       ---------------------------------------------------------
C       COMPUTE THE LAST ROW OF A. I MOVES ACROSS THE COLUMNS,
C       SUMS OVER THE N VALUES.
C       ---------------------------------------------------------
        DO 50 I = 2, MEP1
          A(MEP1,I) = 0.0
          DO 40 J = 1, N
            A(MEP1,I) = A(MEP1,I) + XN(J)
            XN(J) = XN(J)*X1(J)
40        CONTINUE
50      CONTINUE
```

```
C      -----------------------------------------------------------
C      NOW FILL IN THE REST OF THE A MATRIX. I MOVES DOWN THE
C      ROWS, J MOVES ACROSS THE COLUMNS.
C      -----------------------------------------------------------
       DO 70 J = 2, MEP1
         DO 60 I = 1, MF
           A(I,J) = A(I+1,J-1)
60     CONTINUE
70   CONTINUE
C
C      -----------------------------------------------------------
C      NOW CALL A SUBROUTINE TO SOLVE THE SYSTEM. DO THIS FOR
C      EACH DEGREE FROM MS TO MF. GET THE LU DECOMPOSITION OF C      A
C      -----------------------------------------------------------
       CALL LUDCMQ(A,MEP1,10)
C      -----------------------------------------------------------
C      RESET THE R.H.S. INTO C. WE NEED TO DO THIS FOR EACH
C      DEGREE.
C      -----------------------------------------------------------
       MSP1 = MS + 1
       DO 95 I = MSP1, MEP1
         DO 90 J = 1, I
           C(J) = A(J, MEP2)
90     CONTINUE
         CALL SOLNQ(A,C,I,10)
         IM1 = I - 1
C      -----------------------------------------------------------
C      NOW WRITE OUT THE COEFFICIENTS OF THE LEAST SQUARE
C      POLYNOMIAL
C      -----------------------------------------------------------
       PRINT 202, IM1, (C(J),J=1,I)
C      -----------------------------------------------------------
C      COMPUTE AND PRINT THE VALUE OF BETA = SUM OF DEV
C      SQUARED /(N - M - 1).
C      -----------------------------------------------------------
       BETA = 0.0
       DO 94 IPT = 1, N
         SUM = 0.0
         DO 93 ICOEF = 2, I
           JCOEF = I - ICOEF + 2
           SUM = (SUM + C(JCOEF))*X1(IPT)
93     CONTINUE
         SUM = SUM + C(1)
         BETA = BETA + (Y1(IPT) - SUM)**2
94   CONTINUE
       BETA = BETA/FLOAT(N-I)
       PRINT 203, BETA
95   CONTINUE
200    FORMAT(///'DEGREE OF POLYNOMIAL CANNOT EXCEED N - 1.',/
     &   'REQUESTED MAXIMUM DEGREE TOO LARGE -','REDUCED TO',I3)
```

```fortran
201     FORMAT(1X,9F8.2)
202     FORMAT(/'FOR DEGREE OF ',I2,' COEFFICIENTS ARE'//
     &       ' ',5X,6F13.3)
203     FORMAT(9X,' BETA IS ',F10.5//)
        RETURN
        END
C
        SUBROUTINE LUDCMQ(A,N,NDIM)
C       ----------------------------------------------------------------
C       SUBROUTINE LUDCMQ: THIS SUBROUTINE FORMS THE LU
C       EQUIVALENT OF THE SQUARE COEFFICIENT MATRIX A. THE LU, C          N
COMPACT FORM, IS RETURNED IN THE A MATRIX SPACE.
C       THE UPPER TRIANGULAR MATRIX U HAS ONES ON ITS
C       DIAGONAL - THESE VALUES ARE NOT INCLUDED IN THE
C       RESULT.
C       ----------------------------------------------------------------
        DOUBLE PRECISION A(NDIM,NDIM), SUM
        INTEGER N, NDIM, I, J, JM1, IM1, K
C       ----------------------------------------------------------------
        DO 30 I = 1, N
        DO 30 J = 2, N
          SUM = 0.0
          IF (J.LE.I) THEN
            JM1 = J - 1
            DO 10 K = 1, JM1
              SUM = SUM + A(I,K)*A(K,J)
10        CONTINUE
            A(I,J) = A(I,J) - SUM
          ELSE
            IM1 = I - 1
            IF (IM1.NE.0) THEN
              DO 20 K = 1, IM1
                SUM = SUM + A(I,K)*A(K,J)
20        CONTINUE
            ENDIF
C       ----------------------------------------------------------------
C       TEST FOR SMALL VALUE ON THE DIAGONAL
25          IF (ABS(A(I,I)).LT.1.0E-10) THEN
              PRINT 100, I
              RETURN
            ELSE
              A(I,J) = (A(I,J) - SUM)/A(I,I)
            ENDIF
          ENDIF
30      CONTINUE
        RETURN
100   FORMAT(' REDUCTION NOT COMPLETED BECAUSE SMALL
     &            VALUE', 'FOUND FOR DIVISOR IN ROW', I3)
        END
        SUBROUTINE SOLNQ(A,B,N,NDIM)
```

```fortran
C       ------------------------------------------------------------
C       SUBROUTINE SOLNQ: THIS SUBROUTINE FINDS THE SOLUTION
C       TO A SET OF N LINEAR EQUATIONS THAT CORRESPONDS TO THE
C       RIGHT HAND SIDE VECTOR B. THE A MATRIX IS THE LU
C       DECOMPOSITION EQUIVALENT TO THE COEFFICIENT MATRIX OF
C       THE ORIGINAL EQUATIONS, AS PRODUCED BY LUDCMQ. THE
C       SOLUTION VECTOR IS RETURNED IN THE B VECTOR.
C       ------------------------------------------------------------
        DOUBLE PRECISION A(NDIM,NDIM), B(NDIM), SUM
        INTEGER N, NDIM, I, IM1, K, J, NMJP1, NMJP2
C       ------------------------------------------------------------
C       DO THE REDUCTION STEP
        B(1) = B(1)/A(1,1)
        DO 20 I = 2, N
          IM1 = I - 1
          SUM = 0.0
          DO 10 K = 1, IM1
            SUM = SUM + A(I,K)*B(K)
10        CONTINUE
          B(I) = (B(I) - SUM)/A(I,I)
20      CONTINUE
C       ------------------------------------------------------------
C       NOW WE ARE READY FOR BACK SUBSTITUTION. REMEMBER
C       THAT THE ELEMENTS OF U ON THE DIAGONAL ARE ALL ONE.
C       ------------------------------------------------------------
        DO 40 J = 2, N
          NMJP2 = N - J + 2
          NMJP1 = N - J + 1
          SUM = 0.0
          DO 30 K = NMJP2, N
            SUM = SUM + A(NMJP1,K)*B(K)
30        CONTINUE
          B(NMJP1) = B(NMJP1) - SUM
40      CONTINUE
        RETURN
        END
C
        SUBROUTINE RESET(A,MEP1,MEP2)
C       ------------------------------------------------------------
C       THIS SUBROUTINE RESETS A, XN AND C MATRICES TO ZERO
C       ------------------------------------------------------------
        DOUBLE PRECISION A(10,10)
        DO 20 I = 1, MEP1
          DO 10 J = 1, MEP2
            A(I,J) = 0.0
10        CONTINUE
20      CONTINUE
C
        RETURN
        END
```

APPENDIX C.2 - HOW TO USE THE MULTIPLE-BUBBLE SEPARATION AND COMPLETION PROGRAM

This program determines cusp locations and completes overlapped parts of multiple bubbles. The input data files are TOP.DAT and BOTTOM.DAT. TOP.DAT is the data file for top edge point coordinates, and BOTTOM.DAT is the data file for bottom edge point coordinates. These files are obtained by running the Image Processing Program. The output files of the Multiple-Bubble Separation and Completion Program are DERIV.DAT, LEFT.DAT, and RIGHT.DAT. DERIV.DAT is the data file for first and second derivatives for each edge point. LEFT.DAT is the data file for the coordinates of extension points from left. RIGHT.DAT is the data file for the coordinates of extension points from right.

```
C      program: bubble calculation
c         by  L. (Winston ) Zhang
c         University of Illinois at Urbana-Champaign
c         August, 1992
c
       common /blk1/ no(20), area(20), dc(20), rmaj(20), rmin(20),      angl(20)
       common /blk2/ numb, pie, rlim
       common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
       common /blk4/ vsphr(20), velps(20)
       common /blk5/ densv, hfg, tsat, film
       common /blk6/ qtsphr, qtelps
       common /blk7/ qsphr(20), qelps(20)
       common /blk8/ areaav, dmajav, dminav, asurav, velpsav, qelpsav
c
       character*80 film
c
       open (unit=11, file='cal1.dat', status='old')
       open (unit=12, file='nistout1.dat')
c
       pie = acos(-1.0)
       print*, 'Number of bubbles: '
       read*, numb
       print*, 'The limit value for optical distortion is: '
       read*, rlim
       print*, 'The film type is:  '
       read*, film
       print*, 'The saturation temperature is: '
       read*, tsat
c
       call input
       call opcorrect
       call surarea
       call bubvol
       call latent
       call average
       call output
c
       stop
       end
c
       subroutine input
c      --------------------------------------------------------------
c            subroutine input data
c      This subroutine inputs the data file.
c      --------------------------------------------------------------
       common /blk1/ no(20), area(20), dc(20), rmaj(20), rmin(20), angl(20)
```

83

```fortran
        common /blk2/ numb, pie, rlim
c
        do 10 i = 1, numb
          read(11,*) no(i), area(i), dc(i), rmaj(i), rmin(i), angl(i)
10      continue
        return
        end
c
        subroutine opcorrect
c       -----------------------------------------------------------
c               subroutine optical distortion correction
c       This subroutine is used to correct optical distortion
c       -----------------------------------------------------------
        common /blk1/ no(20), area(20), dc(20), rmaj(20), rmin(20), angl(20)
        common /blk2/ numb, pie, rlim
        common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
c
        do 10 i = 1, numb
         if(dc(i).le.rlim) then
           dst = 0.000885*(dc(i))**2 - 0.0014*dc(i) + 0.812898
         else
         dst = -0.85066*exp(6.352297*dc(i) - 36.4274) + 0.00814*dc(i) + 0.789038
         endif
         areaa(i) = area(i)*dst
         a = (rmaj(i)*sin(angl(i)))**2 + (rmin(i)*cos(angl(i)))**2
         b = 2.0/dst*((rmin(i))**2 - (rmaj(i))**2)*sin(angl(i))*cos(angl(i))
         c = 1.0/dst**2*((rmaj(i)*cos(angl(i)))**2
     &        + (rmin(i)*sin(angl(i)))**2)
         if(a.eq.c) then
           angl(i) = pie/2.0
         else
           angl(i) = 1.0/2.0*atan(b/(a-c))
         endif
         rmaja(i) = rmaj(i)*rmin(i)/sqrt(a*(cos(angl(i)))**2
     &         + b*sin(angl(i))*cos(angl(i)) + c*(sin(angl(i)))**2)
         rmina(i) = rmaj(i)*rmin(i)/sqrt(a*(sin(angl(i)))**2
     &         - b*sin(angl(i))*cos(angl(i)) + c*(cos(angl(i)))**2)
10      continue
        return
        end
c
        subroutine surarea
c       -----------------------------------------------------------
c          subroutine surface area
c       This subroutine calculates the surface area of each bubble
c       assuming an ellipsoidal shape.
c       -----------------------------------------------------------
        common /blk2/ numb, pie, rlim
        common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
c
```

```
          do 30 i = 1, numb
            if(rmaja(i).le.rmina(i)) then
c             assume that rmaja and rmina are very close and the
c               bubble is spherical
              asur(i) = 4.0*pie*(rmina(i))**2
            else
c             surface is an ellipsoid: rmaja is the major radius
              r = rmina(i)/rmaja(i)
              asur(i) = 2.0*pie*rmina(i)*rmaja(i)*(r
     &                    + acos(r)/sqrt(1.0 - r**2))
            endif
30        continue
          return
          end
c
          subroutine bubvol
c         ----------------------------------------------------------------
c                    subroutine bubble volume
c         This subroutine is used to calculate the volume of each bubble for
c         both spherical and elliptical shapes.
c         ----------------------------------------------------------------
          common /blk2/ numb, pie, rlim
          common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
          common /blk4/ vsphr(20), velps(20)
c
          do 10 i = 1, numb
            vsphr(i) = 4.0*pie*(rmaja(i)*rmina(i))**1.5/3.0
            velps(i) = 4.0*pie*rmaja(i)*(rmina(i))**2/3.0
10        continue
          return
          end
c
          subroutine latent
c         ----------------------------------------------------------------
c                        subroutine latent
c         This subroutine calculates the amount of energy used to generate
c         the visualized bubble. Calculations are done assuming both
c         spherical and elliptical bubble shapes.
c         ----------------------------------------------------------------
          common /blk2/ numb, pie, rlim
          common /blk4/ vsphr(20), velps(20)
          common /blk5/ densv, hfg, tsat, film
          common /blk6/ qtsphr, qtelps
          common /blk7/ qsphr(20), qelps(20)
c
          character*80 film
c
          if(film.eq.'R123') then
          densv = 0.0026795152*tsat**2 - 1.379728248*tsat + 179.06845
            hfg = -0.98076923*tsat**2 + 187.431319*tsat + 202939.835
```

```fortran
      endif
      if(film.eq.'R11') then
         densv = 0.0025982882*tsat**2 - 1.334749534*tsat + 173.0516
         hfg = -0.651437929*tsat**2 + 3.90200809*tsat + 238129.844
      endif
c       qtsphr and qtelps is the total energy of all numb bubbles for
c       spherical and elliptical bubbles, respectively.
      qtsphr = 0.0
      qtelps = 0.0
c       qsphr is the heat required for the spherical bubble
c       qelps is the heat required for the elliptical bubble.
      do 10 i = 1, numb
         qsphr(i) = vsphr(i)*densv*hfg/1.0E+09
         qelps(i) = velps(i)*densv*hfg/1.0E+09
         qtsphr = qtsphr + qsphr(i)
         qtelps = qtelps + qelps(i)
10    continue
      return
      end
c
      subroutine average
c       ----------------------------------------------------------------
c       This subroutine calculates the averages of area, major diameter, minor diameter, surface
area, volume of ellipsoid and heat flux.
c       ----------------------------------------------------------------
      common /blk2/ numb, pie, rlim
      common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
      common /blk4/ vsphr(20), velps(20)
      common /blk6/ qtsphr, qtelps
      common /blk8/ areaav, dmajav, dminav, asurav, velpsav, qelpsav
c
      asum = 0.0
      dmajsum = 0.0
      dminsum = 0.0
      ssum = 0.0
      vsum = 0.0
      do 10 i = 1, numb
         asum = asum + areaa(i)
         dmajsum = dmajsum + rmaja(i)*2.0
         dminsum = dminsum + rmina(i)*2.0
         ssum = ssum + asur(i)
         vsum = vsum + velps(i)
10    continue
      div = float(numb)
      areaav = asum/div
      dmajav = dmajsum/div
      dminav = dminsum/float(numb)
      asurav = ssum/float(numb)
      velpsav = vsum/float(numb)
      qelpsav = qtelps/float(numb)
```

```
c
        return
        end
c
        subroutine output
c       -----------------------------------------------------------
c       This subroutine writes the output as data file.
c       -----------------------------------------------------------
        common /blk1/ no(20), area(20), dc(20), rmaj(20), rmin(20), angl(20)
        common /blk2/ numb, pie, rlim
        common /blk3/ areaa(20), rmaja(20), rmina(20), asur(20)
        common /blk4/ vsphr(20), velps(20)
        common /blk5/ densv, hfg, tsat, film
        common /blk6/ qtsphr, qtelps
        common /blk7/ qsphr(20), qelps(20)
        common /blk8/ areaav, dmajav, dminav, asurav, velpsav, qelpsav
c
        character*80 film
        dimension dmaja(20), dmina(20)
c
        write(12,10) film
10      format (//5x, 'Bubbles from film ', a20/)
        write(12,20)
20      format(/'No.', 4x, 'Area', 4x,'Majdia', 4x, 'Mindia', 4x,
     &    'Asurelps', 4x, 'Vsp/Vel', 4x, 'Velps', 4x, 'Qelps')
        write(12,30)
30      format(6x, '(mm^2)', 4x, '(mm)', 6x, '(mm)', 6x, '(mm^2)', 15x,
     &    '(mm^3)', 6x, 'J'/)
        do 50 i = 1, numb
          dmaja(i) = 2.0*rmaja(i)
          dmina(i) = 2.0*rmina(i)
          ratio = vsphr(i)/velps(i)
          write(12,40) no(i), areaa(i), dmaja(i), dmina(i), asur(i),
     &                 ratio, velps(i), qelps(i)
40        format(i2,4x,f6.3,3x,f6.4,4x,f6.4,4x,f8.5,4x,f6.3,3x,f7.5,
     &             3x,e10.4)
50      continue
        write(12,60)
60      format(//5x,'Heat Load of All Bubbles: '/)
        write(12,70) qtsphr
70      format(5x,'Assumed Spherical (J): ', e11.5)
        write(12,80) qtelps
80      format(5x,'Assumed Elliptical (J): ', e11.5)
        write(12,90) densv
90      format(5x,'Density of Vapor (kg/m3): ', f8.5)
        write(12,95) hfg
95      format(5x,'Latent Heat (J/kg): ', f12.2)
        write(12,100) tsat
100     format(5x,'Saturation Temperature (K): ', f8.2)
        write(12,110)
```

```fortran
110     format(//5x,'Averaged Quantities: '/)
        write(12,120) areaav
120     format(5x,'Cross Sectional Area (mm^2): ', f8.5)
        write(12,130) dmajav
130     format(5x,'Major Diameter (mm): ', f8.5)
        write(12,140) dminav
140     format(5x,'Minor Diameter (mm): ', f8.5)
        write(12,150) asurav
150     format(5x,'Surface Area of Ellipsoid (mm^2): ', f8.5)
        write(12,160) velpsav
160     format(5x,'Volume of Ellipsoid (mm^3): ', f8.5)
        write(12,170) qelpsav
170     format(5x,'Required Energy to Make Ellipsoid (J/Bub.): ', e11.5)
        return
        end
```

APPENDIX D.2 - HOW TO USE THE ENERGY CALCULATION PROGRAM

The Energy Calculation Program uses the data from NIH Image for input data. The input data file contains areas, major and minor radii, and angles. Shown in Table C.1 is a typical input data file.

Table C.1 - Input data before correcting for optical distortion

#	Area(mm2)	dc(mm)	Rmaj(mm)	Rmin(mm)	Angle(rad)
1	0.650	0.700	0.645	0.320	0.442
2	0.530	0.720	0.565	0.300	0.012
3	0.800	0.980	0.700	0.365	0.571
4	0.510	1.090	0.430	0.375	2.922
5	0.360	1.910	0.400	0.285	0.036
6	0.910	2.370	0.730	0.395	0.019
7	0.360	4.090	0.405	0.285	0.132

The program then corrects for optical distortion using formulae provided by NIST. It then calculates the surface area and volume assuming the bubbles are elliptical to obtain energy content within each bubble. An output data file corresponding to the above input data file is shown in Table C.2.

Table C.2 - Output data using R123 properties

#	Area (mm2)	Majdia (mm)	Mindia (mm)	Asurelps (mm2)	Vsp/Vel	Velps (mm3)	Qelps (J)
1	0.482	1.2422	0.4925	1.59854	1.588	0.15777	0.1947E-03
2	0.393	1.1300	0.4446	1.31156	1.594	0.11693	0.1443E-03
3	0.592	1.3165	0.5743	1.99559	1.514	0.22738	0.2806E-03
4	0.377	0.8562	0.5572	1.33860	1.240	0.13918	0.1718E-03
5	0.267	0.7998	0.4235	0.91689	1.374	0.07510	0.9270E-04
6	0.681	1.4599	0.5911	2.25946	1.572	0.26707	0.3296E-03
7	0.285	0.8081	0.4521	0.99758	1.337	0.08649	0.1068E-03

Heat Load of All Bubbles:

Assumed Spherical (J): 0.19653E-02
Assumed Elliptical (J): 0.13206E-02
Density of Vapor (kg/m3): 7.29364
Latent Heat (J/kg):   169227.00
Saturation Temperature (K):   304.13

Averaged Quantities:

Cross Sectional Area (mm^2): 0.43950

Major Diameter (mm): 1.08751
Minor Diameter (mm): 0.50504
Surface Area of Ellipsoid (mm^2): 1.48832
Volume of Ellipsoid (mm^3): 0.15284
Required Energy to Make Ellipsoid (J/Bub.): 0.18865E-03