BMEN 6003: Module 1, Echocardiography Lab Weeks 2 and 3 Report

Michael Yang
Graduate Student, Columbia University
New York, United States
my2699@columbia.edu

***Abstract*—This experiment uses a balloon as well as patient-derived measurements for long-axis and short-axis dimensions in order to assess the accuracy of 2D and 3D echocardiography techniques in determining as end-diastolic and end-systolic volume, stroke volume, ejection fraction, and cardiac output. A MATLAB routine was developed to further investigate volume estimations from patient-derived echocardiography data versus provided videos of an unknown patient's heart.**

I. Introduction

As noted in the lectures for this course, the heart has an extremely important role in pumping blood throughout the body. Blood volume mainly exits the Left Ventricle and goes into the body; therefore, blood pressure and pumping within the Left Ventricle and aorta determine the flow of the blood, from highest to least pressure during the ejection phase. Right after the heart is fully pumped, there is the most possible blood volume within the Left Ventricle; this stage is known as End Diastole. The Mitral Valve, which is the valve between the Left Atrium and Left Ventricle, usually opens at this stage. The stage right after blood ejection from the Left Ventricle is known as End Systole, in which blood volume is smallest in the Left Ventricle. Within the clinic, there is a need to understand how much blood is pumped out of th heart in each pumping cycle, or heart beat, known as Cardiac Output. Similarly, cardiologists also use the blood volumes measured during End Diastole and End Systole to measure Stroke Volume.

As mentioned, it is clinically relevant to evaluate the motion, characteristics, and functionality of the heart. In particular, volume measurement of an organ is very important for diagnostics; in the case of the heart, understanding ventricular function can help to detect changes induced by potential cardiac diseases, and thus diagnose them. One such imaging technique used to monitor these points is Diagnostic Ultrasound, a non-invasive, portable, and inexpensive imaging modality. Volume measurements for organs can be collected from traditional 2D ultrasonography by taking three diameter measurements perpendicular to each other, then multiplying them. In order to do this, however, assumptions about the organ's geometry. For imaging the heart using ultrasound, known as echocardiography, we must account for the long, or major, axis and short, or minor, axis dimensions. Traditionally, 2D ultrasonography presents difficulties in properly measuring changes in long-axis dimensions, as mentioned in *Gilja et al.* 1999 **[1]**; however, in 1992, *Zile et al.* was able to demonstrate a simplified echocardiographic method that was able to provide accurate determinations of left ventricular volumes **[2]**. This experiment utilizes the assumptions which were demonstrated in *Zile et al., 1992*, which note that the ratios of cardiac long and short axis dimensions are constant throughout the cardiac cycle; therefore, it is possible to take a single long-axis measurement, while taking multiple instantaneous short-axis measurements.

It can be hypothesized that measurements taken from 3D echocardiography result in more precise volumetric measurements than 2D as noted in *Gilja et al. 1999*, and that evaluation of cardiac function using the Simpson's Rule to measure volume is more accurate to physiological values in literature than by assuming geometries as documented in *Zile et al.*.

This experiment is necessary to understand the difference in volume estimations derived from measurements taken via 2D and 3D echocardiography, as well as understand how to translate those measurements into clinically relevant values, such as ventricular cavity volume, stroke volume and cardiac output. It is also necessary in order to observe the differences in volume measurements obtained from simple geometric assumptions, such as assuming the ventricular cavity is approximately 3/4 of an axisymmetric ellipsoid, versus more complex volume estimates using the Simpson's rule formula. Therefore, the objectives of this study are to:

- Compute and observe differences obtained from 2D and 3D echocardiography-based volume estimation techniques, using a balloon as a phantom.
- Evaluate differences in calculated cardiac function parameters using patient-obtained 2D transthoracic echocardiography images, between calculations made with volumetric assumptions versus MATLAB-based calculations made with the Simpson's rule formula.

II. Materials and Methods

A. *Comparison of 2D and 3D ultrasound volume measurement techniques using Balloons*

First, volume measurements were conducted on a balloon filled with water, which served as a phantom for an actual heart. Unlike the instructions originally intended, this experiment needed to be conducted virtually; therefore, rather than us students guiding the probe and conducting our own measurements, we instead watched a video in which a member of Professor Konofagou's lab, Dr. Rachel Weber, experienced in ultrasound, guided the probe and gathered the measurements instead. The ultrasound machine used in the experiment was a Philips 3-D Ultrasound Scanner, and all measurement tools were from the program used to visualize the ultrasound images used by Dr. Weber. For this experiment, the experiment implemented an assumption that the balloon has an ellipsoidal geometry, with a volume formula given by:

$$V = (\frac{4}{3})\pi \times \frac{L}{2} \times \frac{D_1}{2} \times \frac{D_2}{2} \quad (1)$$

where L is the major axis dimension, and D_1 and D_2 are orthogonal minor axis dimensions from imaging. As mentioned, students had no part in the imaging process due to the unusual online nature of this laboratory; therefore, Dr. Weber was the only one who guided the probe, chose the frames to image and used the caliper function on the Ultrasound machine to dimension the image. On the scanner, both 2-D and 3-D measurements were taken.A meshing tool was used to roughly estimate the size of the balloon in the 3-D view, which was used by the program together with Simpson's rule to estimate a 3D volume.

B. Evaluation of Cardiac Function using Ultrasound

EKG patches were attached to a human patient, in this case Steven, another member of Professor Konofagou's lab. Again, this was conducted virtually, with Dr. Weber attaching the EKG patches and conducting the readings using the ultrasound probe. The patient was rolled onto the right side, with their left arm under head and right arm on the hip. Ultrasound gel was applied to the patches to prevent attenuation of sound waves with air in the skin-probe interface. 2-D echocardiography was conducted on the patient's heart in the parasternal long-axis position to obtain Left ventricular outflow diameter, as well as long-axis and short-axis dimensions at End-Diastole and End-Systole. The main measurements taken from 2D echocardiography in this stage are: End-diastole long-axis, End-systole long-axis, End-diastole short-axis, and End-systole short-axis. Doppler measurements were also taken using the bimodal feature of the ultrasound machine, which aimed to show directionality of blood flow. The velocity time integral, or VTI, was taken at this stage as well as the max velocity V_{max}. From these measurements, End-diastole and End-systole ventricular cavity volume were taken. The key assumption made here was that the ventricular cavity geometry was assumed to be approximately 3/4 of an axisymmetric ellipsoid, with a volume formula:

$$V = \frac{\pi}{6}(D_endo)^2(L_endo) \quad (2)$$

where D_{endo} is the short-axis measurement, taken via papillary muscle short-axis diameter, and L_{endo} is the long-axis measurement. Additionally, the Stroke Volume was calculated using

$$SV = EDV - ESV \quad (3)$$

where EDV and ESV are end-diastolic and end-systolic volume respectively. The ejection fraction, EF, was calculated using the stroke volume and EDV by using the formula

$$EF = \frac{SV}{EDV} \quad (4)$$

and the cardiac output was measured by using that, as well as a given heart-rate of 60 beats per minute, as:

$$CO = SV \times HR \quad (5)$$

as well as a second calculation of CO using the recorded velocity time interval, using:

$$CO = VTI \times HR \times \frac{\pi D_{AD}^2}{4} \quad (6)$$

where D_{AD} is the diameter of the aorta.

Most important to note here is that the students were not able to conduct the ultrasound experiment on patients themselves, and it was all conducted by Dr. Weber over video with the students simply taking notes on the acquired measurements.

C. Evaluation of Cardiac Function using Ultrasound images + Simpson's Rule formula

A user-controlled MATLAB routine was developed which follows a similar procedure to that of the previous section. Within the routine, the user first can input video files of captured 2D ultrasound on a human heart patient. For each video, the routine allows the user to first analyze frame-by-frame; following this, the code instructs the user to choose 1 frame in particular to use that they believe represents most accurately the end-diastole stage of the left ventricle.

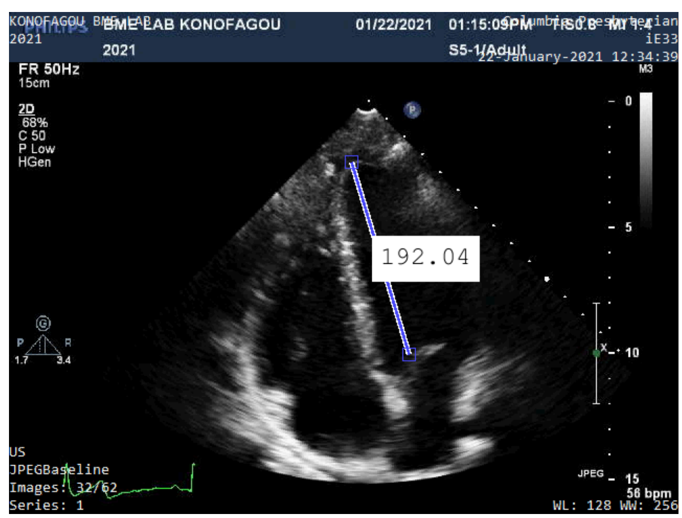

Figure 1. Example of using the MATLAB routine for measuring axis dimensions in a captured frame. The user currently has already decided on and captured a single frame to represent the End-Diastole section of the *Apical 4 Chamber* view, and is currently using the ruler tool in MATLAB image tools (generated automatically during the workflow of

operating the code) to record the distance of long-axis end-diastole in pixels. The code will then automatically convert the pixel measurement into centimeters for further calculation

After the user chooses the desired frame, the program then immediately opens the desired frame as an image, and the user is then able to use the ruler tool to draw the distance for long or short-axis measurements (**Figure 1**).While the ruler measures the distances in units of *pixels*, the code converts measured and entered pixel measurements into centimeters for further calculations. The code then prompts the user to record that measurement, and the code runs through the necessary calculations to return to the user the following characteristics of the heart as derived from their chosen image: end-diastolic volume, end-systolic volume, stroke volume, ejection fraction, and cardiac output. The calculations for EDV and ESV are based on the Simpson's Rule Formula, which is as follows:

$$V = (A_1 + A_2)h + (\frac{A_3 h}{2}) + (\frac{3}{6}) \tag{7}$$

where h = long-axis distance / 3, A_1 = mitral valve short-axis area, A_2 = papillary muscle level area, and A_3 = apex short-axis area. All areas were calculated with the assumption that the geometry is that of a perfect circle. Within this particular experiment, the code was written in such a way to only process the four given videos, which focus on a apical 4 chamber view, mitral, papillary, and apex views. Therefore, the code first started at the apical 4 chamber views and cycled through the four videos as described in the order above.

III. Results

A. Results of Balloon Experiment

Measurements conducted on the Balloon can be found in **Table 1**. As noted, all measurements were conducted by Dr. Weber and the caliper tool on the ultrasound machine over video, so we did not have the opportunity to follow the original instructions and fill + measure the balloon ourselves. 2D measurements were conducted on one long-axis as well as 2 minor axis measurements, with the first measurement perpendicular to the long (major) axis measurement, and the second measured along the orthogonal dimension in which the probe was turned 90 degrees. For 3D measurements, all values were obtained from Dr. Weber and the ultrasound program.

Balloon Results	
Variables	Measured/Calculated Values
L	11
D_1	10.7
D_2	9.7
V_{2D}	538.99
V_{3D}	595.9
V_{True}	640

Table 1. Measurements and Calculations made for the Balloon portion of the experiment. All distance measurements, including L, D_1, D_2, were recorded in cm. The V_3D and V_2D were calculated from the 3D-mesh features of the ultrasound program and the given equation **(1)**, respectively. The V_{True}, or true volume of the balloon, was provided by Dr. Weber. All volume measurements are in mL.

B. Results from 2D and 3D ultrasound on human patient

Measurements and results of evaluating cardiac function using ultrasound on a patient, Steven, is documented in **Table 2**. Again, all measurements were conducted by Dr. Weber for this experiment instead of by us students; therefore, these measurements were simply recorded as they were reported over the video.

Live Patient Results	
Variables	Measured/Calculated Values
$L_{ED,2D}$	8.40
$L_{ES,2D}$	7.14
$D_{Endo,ED}$	5.40
$D_{Endo,ES}$	3.61
Heart Rate	60
Aorta Diameter D_{AD}	2.0
VTI	19.0
$VTI_{continuous}$	23.8
EDV	128.25
ESV	48.7
SV	79.55
EF	0.6203
CO_{HR}	4773
CO_{VTI}	3581
$CO_{VTI,continuous}$	4486

Table 2. Measurements and Calculations made using Steven as the patient, and abovementioned equations (2-6). All distance values, including L and D values as well as D_AD and VTI, are in cm, heart rate is in beats per minute, and CO is in mL/min.

C. Results of MATLAB routine

All calculations were conducted within the MATLAB code, and results for the measurements can be found in **Table 3** and calculations in **Table 4**. As mentioned, the routine used videos that were provided, and each individual image in terms of desired frames for systole and diastole are based on the user's decision; therefore, the results displayed are an example based on the frames chosen by me. For reference, the frames chosen for these results are: **[Apical 4 Chambers]:** 32 and 16, **[Mitral]:** 29 and 16, **[Papillary]:** 28 and 50, and **[Apex]:** 28 and 50.

Video-based MATLAB Results		
Video	Variable	Measured Value (cm)
Apical 4 Chambers	L_{ED}	8.05
	L_{ES}	6.63
Mitral	$S_{ED,M}$	5.25
	$S_{ES,M}$	3.46
Papillary	$S_{ED,P}$	5.65
	$S_{ES,P}$	3.75
Apex	$S_{ED,A}$	4.70
	$S_{ES,A}$	4.10

Table 3. Measurements conducted in MATLAB image analysis using developed routine. Each image has 2 main measurements, conducted using the ruler tool in MATLAB image tools. **4 Channels** is where we obtained long-axis measurements in end-systole and end-diastole, and the other images were used for measuring short-axis during the two stages.

MATLAB Generated Results	
Variables	Calculated Values
EDV	66.0
ESV	159.5
SV	93.5
EF	0.5863
CO_{HR}	5612

Table 4. Calculations that were conducted in MATLAB. All calculations are built-into the MATLAB code, and formulas used are **(3)**, **(4)**, **(5)**, and **(7)**.

Due to the unusual nature of the experiment, as well as the deviation in the procedure from the norm, statistical analysis was not conducted on this data.

IV. Discussion

As noted, the primary objectives of this experiment were divided into two main sections: the analysis of 2D versus 3D ultrasound in accurately estimating cardiac volume, using a balloon as a phantom model, and the differences between geometric assumptions and the use of ultrasound-provided measurements versus MATLAB-generated measurements and calculations in determining cardiac volume as well as other characteristics. In the balloon experiment, 2D measurements were taken using the ultrasound probe. From those, long-axis and 2 short-axis measurements were taken, and using a formula for volume on the geometric assumption that the balloon was an ellipsoid, the volume was calculated to be 595.9 mL. Meanwhile, 3D measurements were conducted based on a mesh view from the 3D view on the scanner, and volume measurements were conducted from the cross-sectional areas generated by the mesh as well as Simpson's rule. Notably, the volume calculated using the 2D readings, 595.5 mL, were closer to the true volume of the balloon, which was 640 mL compared to the 3D measurement of 538.99 mL. This indicates that the 2D measurement, which used a single long-axis measurement and 2 short-axis measurements, as well as an ellipsoidal assumption, was more accurate than a 3D echocardiogram measurement. This is likely due to several factors, including the variability that can come from the user deciding the boundaries for creating the 3D mesh as shown in the procedure by Steven on video. Dr. Weber was responsible for determining the boundaries for 2D analysis, while Steven had to roughly draw an outline of the spheroid shape of the balloon for generating the 3D mesh during the view. Therefore, the human input required for the determining of those boundaries likely had an effect on the volume measurement. This is consistent with results reported in 2011 by *Herberg et al.* **[3]**, who noted that such discrepancies can directly influence the accuracy of 2D and 3D measurements. It could also be possible that the balloon is indeed more ellipsoidal in nature, which means that the geometric estimation is actually valid when determining the volume formula to use; unlike an actual heart chamber, where the geometry can be very complex especially during motion and more so during due to the influence of cardiac diseases, a balloon's simple geometry means that the volume estimations from 2D echocardiography based on those geometry assumptions can be more valid. This means that, in the case of our hypothesis, it can be noted that the simple geometry of a model can actually influence the accuracy of 2D ultrasound versus 3D ultrasound in volume estimation, and testing the accuracy of 2D versus 3D is influenced by the simplicity of the phantom shape in line with the chosen volume estimation formula as well. It would be interesting to test this on phantoms of more difficult geometries and demonstrate if this is a valid assertion in the future.

For the second objective, the main task was to use both the patient-derived values that were given by Dr. Weber and compare them with user-generated values from running a MATLAB routine to choose own frames and axis dimensions. The values for EDV and ESV obtained by Dr. Weber on Steven, an average to well-built adult male, were found to be 128.25 and 48.7, respectively. Meanwhile, the values found in MATLAB on the subject in the videos were 159.5 and 66, which were slightly higher. The values obtained via MATLAB, which utilized the Simpson's Rule formula for volume calculation, were found to be closer to the average EDV/ESVs in adult males found within literature, 153 mL $\pm$ 30 *mL* and 76 $\pm$ 13 *mL*, compared to the values found using an axisymmetrical ellipsoid volume estimation **[4]**. Additionally, the values from Stroke Volume and Ejection Fraction in MATLAB differed as well from calculations obtained from Dr. Weber's measurements; Stroke volume increased in the MATLAB code while ejection fraction decreased. The Cardiac Output was also higher than all values obtained from the various methods in the Patient-Derived values, despite using the same heart rate. Notably, the cardiac output values obtained from using heart rate were more close to the average cardiac output for a body at rest found in literature, 5 L/min, compared to values found using *VTI* **[5]**. Part of this may be due to the more complex geometries of a patient-derived heart, allowing for the Simpson's rule formula to more accurately estimate left ventricular volume than an ellipsoidal estimation. These results did affirm the validity of the MATLAB routine and using the modified Simpson's rule for LV volume estimation. Of note, as in the MATLAB code the user is responsible for choosing frames and determining the measurements using the ruler tool, the results in this experiment are not final values and are subject to change depending on the user's experience with the subject. Additionally, the values obtained from the MATLAB were found to be clinically relevant for a healthy adult male, which indicates that the videos of an unknown patient may also be a healthy adult; therefore, it would be interesting to see how drastically these values differ for diseased hearts.

References

[1] O. H. Gilja, T. Hausken, A. Bernstad, and S. Odegaard, "Measurements of organ volume by ultrasonography," Proc. Instn. Mech. Engrs., vol. 213, Part H., pp. 247–259, 1999.

[2] M.R. Zile, R. Tanaka, J.R. Lindroth, F. Spinale, B. A. Carabello, I. Mirsky, "Left ventricular volume determined echocardiographically by assuming a constant left ventricular epicardial long-axis/short-axis dimension ratio throughout the cardiac cycle, " JACC., vol. 20, no. 4, pp. 986–993, October 1992.

[3] U. Herberg, M. Brand, C. Bernhardt, H. George Trier, J. Breur, "Variables influencing the accuracy of 2-dimensional and real-time 3-dimensional echocardiography for assessment of small volumes, areas, and distances," J. Ultrasound Med., vol. 30, pp. 899–908, March 2011.

[4] P. A. Cain, R. Ahl, E. Hedstrom, M. Ugander, A. Allansdotter-Johnsson, P. Friberg, H. Arheden, "Age and gender specific normal values of left ventricular mass, volume, and funciton for a gradient echo magnetic resonance imaging: a cross sectional study, " BMC Med. Imaging, vol. 9, no. 2, January 2009.

[5] N. Patel, J. Durland, A. N. Makaryus, "Physiology, cardiac index, " StatPearls Publishing, January 2020.

V. APPENDIX

All code for this project can be found in the MATLAB file, $"Assignment_01_YangMichael.m"$. The videos associated with the file are found as $"Apical4chVolunteer.avi"$, $"SA_MitralVolunteer.avi"$, $"SA_PapillaryVolunteer.avi"$, and $"SA_ApexVolunteer.avi"$.

Some main issues with the code that I could not fit into the original document:

- The Code requires user to understand the basic steps of the process, such as first picking a frame, then choosing the distance with a ruler
- The code's dialog box currently only opens AFTER the user views the frames, which means the user actually needs to log or remember the frame number **as well as** the distance they used before they are able to input the value into the dialog box.
- The code currently cycles through all videos at once, with only short intermissions in-between letting the user know that they are moving from one video to the next video, as well as telling them what the next video would be. The code is NOT yet able to have the user input any kind of video on their own.

Other than these points, the code is fully functional and allows the user to only provide input for the frame number and the distance they wish to use for long or short axis measurements; the code then calculates the distance, converts to cm, and then inputs those values into the respective formulas before giving a message box containing the calculated values for EDV, ESV, SV, EF, and CO as mentioned in the MATERIALS AND METHODS section of this report.

For reference, here is a transcript of the code from MATLAB:

$Apical =' Apical4chVolunteer.avi'; Mitral =' SA_MitralVolunteer.avi'; Papilary =' SA_PapilaryVolunteer.avi'; Apex =' SA_ApexVolunteer.avi';$

$Apic_vid = VideoReader(Apical); Mit_vid = VideoReader(Mitral); Pap_vid = VideoReader(Papilary); Apex_vid = VideoReader(Apex);$

$conversion = 1/24;$

$Samples = [1234];$

$fori = Samplesifi == 1play11 = implay(Apical); set(play11.Parent,' Name',' Apical4Chambers-RecordFrameNumberforEndDiastole'); waitfor(play11); prompt1 'EnterFrameNumber :'; dlgtitle =' ChooseFrameforApicalEndDiastole'; answerApicalED = inputdlg(prompt1, dlgtitle); answerApicalED_num = answerApicalED1; ans11 = str2double(answerApicalED_num); this_frame11 = read(Apic_vid, ans11);$

$tool11 = imtool(this_frame11); waitfor(tool11); prompt11 = 'EnterEDlong - axisdistance :'; dlgtitle =' InputMeasuredLong - AxisDistance'; input11 = inputdlg(prompt11, dlgtitle); input11_num = input111; ED_ans = str2double(input11_num); ED_cm = ED_ans * conversion;$

$play12 = implay(Apical); set(play12.Parent,' Name',' Apical4Cha RecordEndSystoleFrame'); waitfor(play12); prompt2 = 'EnterFrameNumber :'; dlgtitle2 =' ChooseFrameforApicalEndSystole'; answerApicalES = inputdlg(prompt2, dlgtitle2); answerApicalES_num = answerApicalES1; ans12 = str2double(answerApicalES_num); this_frame12 = read(Apic_vid, ans12);$

$tool12 = imtool(this_frame12); waitfor(tool12); prompt12 = 'EnterESlong - axisdistance :'; dlgtitle =' InputMeasuredlong - AxisDistance'; input12 = inputdlg(prompt12, dlgtitle); input12_num = input121; ES_ans = str2double(input12_num); ES_cm = ES_ans * conversion;$

$fig = uifigure; selection = uiconfirm(fig,' MovingOntoMitralVideo',' Inter - framenotice');$

$elseifi == 2play21 = implay(Mitral); set(play12.Parent,' Name',' Mitral - RecordEnd-Diastoleframe'); waitfor(play21); prompt3 = 'EnterFrameNumber :'; dlgtitle3 =' ChooseFrameforMitralEndDiastole'; answerMitralED = inputdlg(prompt3, dlgtitle3); answerMitralED_num = answerMitralED1; ans21 = str2double(answerMitralED_num); this_frame21 = read(Mit_vid, ans21);$

$tool21 = imtool(this_frame21); waitfor(tool21); prompt21 = 'EnterEDshort - axisdistance :'; dlgtitle =' InputMeasuredShort - AxisDistance'; input21 = inputdlg(prompt21, dlgtitle); input21_num = input211; SA_ans_MitED = str2double(input21_num); SA_cm_MitED = SA_ans_MitED * conversion;$

$play22 = implay(Mitral); set(play12.Parent,' Name',' Mitral-RecordEnd-Systoleframe'); waitfor(play22); prompt4 =$

$'EnterFrameNumber:'; dlgtitle4 =' ChooseFrameforMitralEndSystole'; answerMitralES = inputdlg(prompt4, dlgtitle4); answerMitralES_{n}um = answerMitralES1; ans22 = str2double(answerMitralES_{n}um); this_{f}rame22 = read(Mit_{v}id, ans22);$

$tool22 = imtool(this_{f}rame22); waitfor(tool22); prompt22 = 'EnterESshort-axisdistance:'; dlgtitle =' InputMeasuredShort - AxisDistance'; input22 = inputdlg(prompt22, dlgtitle); input22_{n}um = input221; SA_{a}ns_{M}itES = str2double(input22_{n}um); SA_{c}m_{M}itES = SA_{a}ns_{M}itES * conversion;$

$fig = uifigure; selection = uiconfirm(fig,' MovingOntoPapilaryVideo',' Inter - framenotice');$

$elseifi == 3play31 = implay(Papilary); set(play12.Parent,' Name',' Papilary-RecordEnd-Diastoleframe'); waitfor(play31); prompt5 = 'EnterFrameNumber:'; dlgtitle5 =' ChooseFrameforPapilaryEndDiastole'; answerPapED = inputdlg(prompt5, dlgtitle5); answerPapED_{n}um = answerPapED1; ans31 = str2double(answerPapED_{n}um); this_{f}rame31 = read(Pap_{v}id, ans31);$

$tool31 = imtool(this_{f}rame31); waitfor(tool31); prompt31 = 'EnterEDshort-axisdistance:'; dlgtitle =' InputMeasuredShort - AxisDistance'; input31 = inputdlg(prompt31, dlgtitle); input31_{n}um = input311; SA_{a}ns_{P}apED = str2double(input31_{n}um); SA_{c}m_{P}apED = SA_{a}ns_{P}apED * conversion;$

$play32 = implay(Papilary); set(play12.Parent,' Name',' Papilary-RecordEnd-Systoleframe'); waitfor(play32); prompt6 = 'EnterFrameNumber:'; dlgtitle6 =' ChooseFrameforPapilaryEndSystole'; answerPapES = inputdlg(prompt6, dlgtitle6); answerPapES_{n}um = answerPapES1; ans32 = str2double(answerPapES_{n}um); this_{f}rame32 = read(Pap_{v}id, ans32);$

$tool32 = imtool(this_{f}rame32); waitfor(tool32); prompt32 = 'EnterESshort-axisdistance:'; dlgtitle =' InputMeasuredShort - AxisDistance'; input32 = inputdlg(prompt32, dlgtitle); input32_{n}um = input321; SA_{a}ns_{P}apES = str2double(input32_{n}um); SA_{c}m_{P}apES = SA_{a}ns_{P}apES * conversion;$

$fig = uifigure; selection = uiconfirm(fig,' MovingOntoApexVideo',' Inter - framenotice');$

$elseifi == 4play41 = implay(Apex); set(play12.Parent,' Name',' Apex - RecordEnd-Diastoleframe'); waitfor(play41); prompt7 = 'EnterFrameNumber:'; dlgtitle7 =' ChooseFrameforApexEndDiastole'; answerApexED = inputdlg(prompt7, dlgtitle7); answerApexED_{n}um = answerApexED1; ans41 = str2double(answerPapED_{n}um); this_{f}rame41 = read(Apex_{v}id, ans41);$

$tool41 = imtool(this_{f}rame41); waitfor(tool41); prompt41 = 'EnterEDshort-axisdistance:'; dlgtitle =' InputMeasuredShort - AxisDistance'; input41 = inputdlg(prompt41, dlgtitle); input41_{n}um = input411; SA_{a}ns_{A}pexED = str2double(input41_{n}um); SA_{c}m_{A}pexED = SA_{a}ns_{A}pexED * conversion;$

$play42 = implay(Apex); set(play12.Parent,' Name',' Apex-RecordEnd-Systoleframe'); waitfor(play42); prompt8 = 'EnterFrameNumber:'; dlgtitle8 =' ChooseFrameforApexEndSystole'; answerApexES = inputdlg(prompt8, dlgtitle8); answerApexES_{n}um = answerApexES1; ans42 = str2double(answerPapES_{n}um); this_{f}rame42 = read(Apex_{v}id, ans42);$

$tool42 = imtool(this_{f}rame42); waitfor(tool42); prompt42 = 'EnterESshort-axisdistance:'; dlgtitle =' InputMeasuredShort - AxisDistance'; input42 = inputdlg(prompt42, dlgtitle); input42_{n}um = input421; SA_{a}ns_{A}pexES = str2double(input42_{n}um); SA_{c}m_{A}pexES = SA_{a}ns_{A}pexES * conversion;$

$fig = uifigure; selection = uiconfirm(fig,' MovingToDisplayValues',' Notice');$

end end

$A1_{E}D = pi * (SA_{c}m_{M}itED/2)^2; A2_{E}D = pi * (SA_{c}m_{P}apED/2)^2; A3_{E}D = pi * (SA_{c}m_{A}pexED/2)^2; h_{E}D = (ED_{c}m/3); EDV = ((A1_{E}D + A2_{E}D) * h_{E}D) + ((A3_{E}D * h_{E}D)/2) + ((pi * (h_{E}D^3))/6);$

$A1_{ES} = pi * (SA_{cmMitES}/2)^2; A2_{ES} = pi * (SA_{cmPapES}/2)^2; A3_{ES} = pi * (SA_{cmApexES}/2)^2; h_{ES} = (ES_{cm}/3); ESV = ((A1_{ES} + A2_{ES}) * h_{ES}) + ((A3_{ES} * h_{ES})/2) + ((pi * (h_{ES}^3))/6);$

$SV = EDV - ESV;$

$EF = SV/EDV;$

$HeartRate = 60; CO = SV * HeartRate;$

$formatspec1 =' TheEndDiastolicVentricularCavityVolumeis : msg1 = sprintf(formatspec1, EDV); formatspec2 =' TheEndSystolicVentricularCavityVolumeis : msg2 = sprintf(formatspec2, ESV); formatspec3 =' TheStrokeVolumeis : msg3 = sprintf(formatspec3, SV); formatspec4 =' TheEjectionFractionis : msg4 = sprintf(formatspec4, EF); formatspec5 =' TheCardiacOutputinmL/mingivenHR = 60, is : msg5 = sprintf(formatspec5, CO); f = msgbox(msg1, msg2, msg3, msg4, msg5,' CalculatedValues');$