AUTOMATIC IDENTIFICATION OF THE SCAPULAR BORDER TO INCREASE THE EFFICIENCY OF DATA PROCESSING FOR THE FREEHAND THREE-DIMENSIONAL ULTRASOUND SYSTEM

by

Imaobong Aniema Udofa

B.S. in Computer Engineering, University of Maryland, Baltimore County, 2013

Submitted to the Graduate Faculty of

The Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Imaobong Aniema Udofa

It was defended on

April 15, 2015

and approved by

Alicia Koontz, Ph.D.
Associate Professor, Department of Rehabilitation Science and Technology

Lynn Worobey, Ph.D.
Research Scientist, Department of Physical Medicine and Rehabilitation

Thesis Advisor: Michael Boninger, M.D.
Chair and Professor, Department of Physical Medicine and Rehabilitation

AUTOMATIC IDENTIFICATION OF THE SCAPULAR BORDER TO INCREASE THE EFFICIENCY OF DATA PROCESSING FOR THE FREEHAND THREE-DIMENSIONAL ULTRASOUND SYSTEM

Imaobong Aniema Udofa, M.S.

University of Pittsburgh, 2015

The ability to visualize the scapula in three dimensions (3D) is necessary for the evaluation of scapular movement. The scapula plays an important role in upper extremity function as it provides a stable base for shoulder movement and enables optimal shoulder complex function. We previously developed a custom freehand ultrasound (FUS) system for purposes of evaluating bone movement, which is a relatively unexplored application as it pertains to shoulder biomechanics. Our system was developed to create a reconstructed scapular border in 3D space, from points of interest in two-dimensional (2D) ultrasound images, and determine scapular rotations. We found high reliability in evaluating scapular kinematics in static postures with our 3D FUS system. However, we are currently limited to manual detection of the scapular border in the ultrasound images, which is very time consuming. Steps are needed to enhance the FUS system to include automatic detection and increase efficiency. For this study, we have developed a program, capable of automatically identifying and tracking the scapula in 2D ultrasound images, to be integrated into our 3D FUS system. Selected coordinates identified as the scapular border by our automated program were compared to previous manual selections to validate its accuracy and reliability. Using intraclass correlation coefficients, we found substantial to excellent inter-rater reliability (agreement between the automated and manual point selections). The semi-automated point selection program reduces the data processing time required for

identification of the spine and medial border of the scapula in our ultrasound images by over 50%. Our results suggest that this proposed program is a viable method for automatically identifying and tracking the scapular border in 2D ultrasound images. Further study on image pre-processing prior to future application of this automated program should be conducted to further improve the accuracy of our algorithm. In conclusion, point selection is necessary for 3D reconstruction of the scapular border and this automation ultimately enhances our FUS system by increasing the efficiency of our point selection process. Access to 3D scapular models plays several roles ranging from detection of shoulder pathologies to assessing the effectiveness of interventions or preventative measures for shoulder injuries.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 SHOULDER PATHOLOGY: PAIN AND INJURY

Shoulder pain is the third most common musculoskeletal complaint in the general population, with an incidence of 6.6 to 25 cases per 1000 patients[1, 2]. Shoulder pain can affect up to one-fourth of the population[3], depending on age and risk factors, with a peak incidence in the fourth to sixth decades[1]. Shoulder pathology, pain, or injury can decrease the quality of life of an individual as it may negatively impact their independence, occupational status, and/or mobility.

The most frequently encountered shoulder disorders include subacromial impingement, rotator cuff pathology, tendinopathy, or tears, as well as glenohumeral joint instability and adhesive capsulitis[1, 4]. With the exception of adhesive capsulitis, the majority of these shoulder impairments are related to occupational or athletic activities that involve frequent use of the arm at, or above, shoulder level. The point prevalence of shoulder pain in certain sports or occupations can reach 40% or higher[4-7]. Similarly, the repetitive nature and high forces associated with activities of daily living for manual wheelchair users (MWUs) expose them to increased risk of shoulder pain, dysfunction, and pathology. The prevalence of shoulder pain among wheelchair users is higher than that of the general population with about 31-73% encountering shoulder complications[8, 9].

Ultimately, there must be an understanding of the normal biomechanics to understand the pathomechanics of injury or dysfunction[10, 11]. The kinematics of the shoulder joint involves coordinated movements between the clavicle, scapula, and humerus[12]. The clavicle connects the scapula to the central portion of the body to allow the translations (upward/downward and retraction/protraction) around the thorax to occur. The scapula is responsible for providing synchronous scapular rotation during humeral motion and providing a stable base for glenohumeral mobility and rotator cuff activation. The coordinated movement between the scapula and humerus, scapulohumeral rhythm, is necessary for efficient arm motion and to maximize joint stability[11, 13]. If the scapula is unable to perform its stabilization role, shoulder complex function is adversely affected, which may predispose an individual to shoulder injury[11].

1.1.1 Scapular Position and Motion

Abnormal scapular position and motion are found in 68% to 100% of patients with shoulder injuries[14]. Alterations in either scapular motion or position have been found to decrease linear measures of the subacromial space, increase impingement symptoms, decrease rotator cuff strength, increase strain on the anterior glenohumeral ligaments and increase the risk of internal impingement[15]. The evidence of altered scapular kinematics in a variety of shoulder pathologies emphasizes the significance of observing scapular position and dynamic motion. In addition, research states that there is a direct correlation between increased knowledge regarding the role of the scapula and the development of improved evaluation and treatment approaches to scapular dyskinesis[11]. (It is important to note that ‘scapular dyskinesis’ has become the accepted term for altered scapular motion or position[15-17].)

While scapular dyskinesis has been found in relation to a variety of shoulder pathologies, the exact relationship between dyskinesis and clinical pathology remains unclear. In cases other than nerve injury (where dyskinesis is the result of the injury), it is unknown whether scapular dyskinesis is the cause that predisposes the arm to injury or the result of injury that increases dysfunction[11, 15]. Scapular dyskinesis is generally characterized by a lack of both upward rotation and posterior tilting, as well as increased internal rotation of the scapula. However, some methods of evaluating scapular dyskinesis consist of a form of observation in only a single plane[11]. Methods of clinical assessment for scapular dyskinesis that consider only a single plane have proven to be insufficient due to the anatomy of the scapula. The scapula is a flat bone, with extensive overlying soft-tissue, that glides under skin and muscle. The scapula allows for 3 rotational movements (upward/downward, internal/external, anterior/posterior tilt) and 2 translations (upward/downward, retraction/protraction) [13, 16, 18]. Techniques used to evaluate or measure scapular position and motion must take into account the three-dimensional (3D) nature of the scapula.

1.2 FREEHAND THREE-DIMENSIONAL ULTRASOUND

Methods to evaluate scapular movement are continuously being developed and improved. Existing methods include, but are not limited to, bone pins combined with motion capture, magnetic resonance imaging (MRI), radiography (i.e. X-ray), digital fluoroscopy, and skin-based trackers. Unfortunately, none of these methods are without their share of limitations. Insertion of bone pins into the scapula, with added motion capture, is a highly accurate method but extremely invasive and therefore not feasible for routine evaluation. While an MRI is a non-invasive alternative, it is expensive and does not allow for evaluation of dynamic motion. X-rays and fluoroscopy allow evaluation of movement but are often limited to one-dimension and expose individuals to radiation. Lastly, skin-based systems are often not accurate at high angles of humeral elevation[19, 20].

Freehand three-dimensional ultrasound (3D FUS) combines ultrasound and motion tracking and has the potential to overcome the aforementioned limitations. FUS is non-invasive, less expensive, and more readily available in most clinical settings. 3D FUS can also be applied repeatedly to monitor development, without time constraints and without concern for exposure to radiation. Additionally, 3D FUS allows for direction visualization of the bone, evaluation of dynamic motion, and enables 3D reconstruction of scapular movement. In a previous study, Worobey[20] evaluated the reliability of using 3D FUS to measure the three scapular rotations and found substantial reliability with intraclass correlation coefficients (ICCs) ranging from 0.62 to 0.95. Worobey's results agreed with the pattern of movement found in other studies: during humeral elevation, the scapula was found to move toward a more externally rotated, upwardly rotated, posteriorly titled position.

1.3 PREVIOUS RESEARCH

1.3.1 Manual Detection of the Scapular Border

In the aforementioned reliability study by Worobey[20], there was also a goal to establish the reliability of the manual point selection method used to identify the scapular border in 2D ultrasound images. The spine or medial border of the scapula was manually identified in individual frames of multiple ultrasound videos. Within each image, the scapular border was manually identified as an x-y coordinate. The study reported excellent inter- and intrarater reliabilities ranging from 0.975 to 0.995. However, manual detection of the scapular border is very time consuming and therefore decreases the efficiency of the overall 3D FUS system as a potential clinical tool.

1.3.2 Image Processing Background Literature

Ultrasound image processing is a growing and essential field of research supported by the increasing need and use of 2D - 3D ultrasound imaging modalities in clinical settings. Various challenges exist for ultrasound image analysis due to the inherent artifacts of ultrasound imaging. These artifacts include, but are not limited to, significant amounts of speckle or noise in images[21], image dropouts, false borders[22], and the varying nature of ultrasound images[23]. As a result, there are various techniques used for manual and automated ultrasound image processing algorithms. Common techniques include the application of filters in conjunction with speckle suppression or noise removal, as well as edge or boundary detection, and image segmentation or enhancement methods. There are also well accepted assumptions used in the implementation of these techniques for ultrasound image processing. These assumptions typically concern the appearance of a region, or its boundary, as regions of interest may be characterized by a strong gradient[23, 24], tend to have high intensities[23, 25], and are likely to appear brighter than their surroundings[23, 26]. These assumptions hold true for our ultrasound images with the scapular border often having a higher intensity gradient than its surroundings.

The techniques that are most relevant to our scapular ultrasound images and proposed automation algorithm are edge or boundary detection and image segmentation. Chai, et al. defines edge detection as a process to identify the abrupt changes in pixel intensity or the pixels that characterize the boundaries of objects in an image[27]. Image segmentation is also used to identify boundaries in images as it consists of a process of dividing an image into multiple sets of pixels and assigning labels to each pixel to create a simplified representation that is easier to analyze[28]. Our proposed point selection algorithm utilizes similar concepts as we create, analyze and compare sets of pixels from within a larger region of interest and assign new pixel intensity values based on our maximum likelihood factors in order to identify the boundary region of the scapula. Automated and semi-automated algorithms have been proposed in previous literature for detection of various anatomical structures in ultrasound images but to our knowledge, there is not an existing, similar or alternative, algorithm for scapular border identification. However, there are methods that include similar foundational steps such as interactive identification of the region of interest, learning of border properties[22], displacement tracking[29], and weighting of pixels[30].

1.4 RESEARCH GOALS

The work of this thesis is to build upon the freehand three-dimensional ultrasound system, developed in the previous study by Worobey[20], by increasing its overall efficiency. The main objective for this study was to automate the identification of the scapular border in the ultrasound images. A semi-automated point selection program was developed to take the place of the current manual point selection method. In chapter 2 we describe the characteristics of the ultrasound images and points picked by the manual raters that were assessed to determine the appropriate variables to optimize for increased accuracy of the automated program. We also conducted reliability analyses with the manual point selections serving as the gold standard. Chapter 3 presents the findings of the various analyses conducted on ultrasound videos for trials from a subsample of healthy individuals and wheelchair users, with imaging conducted with participants arm at rest, in 3 different planes of static arm elevation, and during dynamic humeral elevation in the scapular plane. Chapter 4 evaluates these findings and assesses the overall reliability of the automated point selection program developed.

2.0 METHODS

As this study builds upon the Freehand Three-Dimensional Ultrasound system presented in Worobey, et al., 2014[20], only a brief summary will be included for the methods that were kept the same.

2.1 PARTICIPANTS

This study included manual wheelchair users (MWUs) and able-bodied individuals. Subjects were eligible to participate in this study if they were over the age of 18, spoke English, and able to raise their arm above their head. To be eligible for the study MWUs also had to use a wheelchair as their primary means of mobility (>80% of mobility). Subjects were excluded from this study if they had a history of fractures or dislocations in the shoulder from which they had not fully recovered, had upper extremity impairment, weakness or spasticity that prevented smooth movement, or if they could not complete reach tasks while seated with support straps around the trunk. Testing occurred both at the National Veteran's Wheelchair Games and the Human Engineering Research Laboratories. This study was approved by the local Institutional Review Board. Consent of each subject was obtained prior to data collection.

2.2 INSTRUMENTATION

Ultrasound imaging was completed using a Philips HD11XE ultrasound machine (Philips Medical Systems; Bothell, Washington), an Epiphan Frame Grabber (Epiphan Systems; Ottawa, Ontario, Canada), and a custom orthogonal attachment, equipped with Vicon markers, that was fitted to the ultrasound probe. Movement was recorded using Vicon Nexus software (Vicon Motion Systems) and 10 Vicon cameras (Vicon Motion Systems). For a full description of the freehand 3D ultrasound system please refer to Worobey, 2014[20].

2.3 TESTING SETUP

A testing chair was designed to isolate dominant shoulder scapular movement. Adjustable straps were used to hold the contralateral shoulder in place and minimize trunk movement. The testing chair was also equipped with an angle-adjustable guide bar to help ensure consistency during arm elevation trials. Vicon markers were placed on the participant's trunk and dominant arm. Marker placement followed ISB standards[31] to include the following bony landmarks: cervical 7 (C7), thoracic 8 (T8), anterior sternoclavicular (sternum), processus xiphoideus (xiphoid), anterior acromioclavicular (acromion), lateral epicondyle, and medial epicondyle. A triad of markers was also placed on the upper arm in case of marker dropout.

2.4 DATA COLLECTION

2.4.1 Ultrasound Imaging

All ultrasound imaging was performed by one operator. The depth of the ultrasound imaging was set to 4cm for all participants. For static trials, participants were imaged in each of the four positions of interest which consisted of the arm by the participant's side at rest and humeral elevation to 90° in the sagittal, frontal, and scapular planes (30° anterior to the frontal plane). The participant held each position for 1 minute during scanning. The ultrasound probe was moved slowly back and forth along the spine of the scapula (~20 sec), then up and down along the medial border of the scapula (~40 sec). Worobey[20] states that less time was spent on the spine of the scapula because it is more superficial and easily visualized. Participants were given a 2 minute rest period between all trials.

For the dynamic trial, a marker was placed on the guide bar at 90° of humeral elevation to maintain consistency. Participants were instructed to raise their dominant arm to the marker over a 5 second period then lower their arm to the resting position over an additional 5 second period. Participants completed 20 repetitions while the ultrasound probe was held at a new location along the spine and medial border of the scapula for each repetition. A 10 second rest period was provided between repetitions to prevent fatigue.

2.5 DATA PROCESSING

Ultrasound videos were read into a custom Matlab (The MathWorks; Natick, Massachusetts) program and written to individual image files. For Worobey's study, the scapular border was manually identified by a human rater as an x-y coordinate in each US image (**Figure 1**). The spine of the scapula was identified as the most posterior edge while the medial border was identified as the most medial edge in the image. For each set of images, frames were analyzed in sequence to allow tracking of the scapula from one image to the next and prevent misidentification of the scapular border[20]. The manually identified data points from Worobey's study served as the 'gold standard' in this study as they were compared to the corresponding points that the automated program selected as the scapular border.

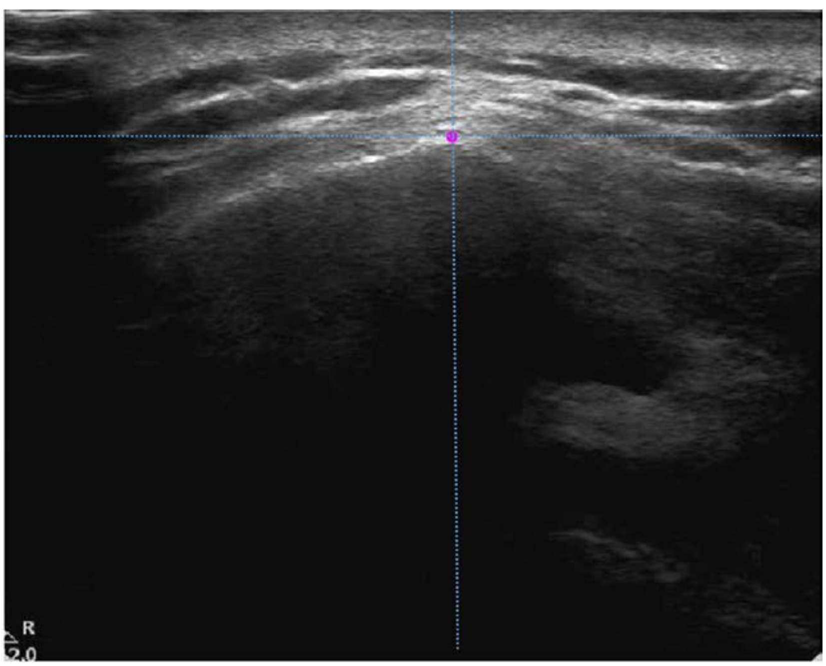

Figure 1. Identification of the Scapular Border as XY coordinate in Ultrasound Image

2.5.1 Pre-Processing

Data pre-processing was performed on a subsample of the subjects included in this study. Pre-processing was conducted to gain a better understanding of the primary features of our ultrasound images and determine how to optimize these variables and incorporate their parameters in the automated program. A custom Matlab program (Appendix A) was written to assess features of each image from each trial used for preprocessing. Each image and the manual point picked (x-y coordinate) were read into the custom program. If the manual point picked signified that the manual (human) rater did not detect an identifiable scapular border in that image, then the program calculated and recorded the total image pixel intensity and proceeded to the next image. If there was an identifiable scapular border detected, as determined by the manual rater, the program calculated the total image pixel intensity, the pixel intensity of the manually selected point, and the pixel intensity of the point 30 pixels below the identified scapular border. The program then uses only the images that had an identifiable scapular border to perform an analysis of the frame to frame distances between coordinates picked to represent the scapular border. The 'total image intensity' is the only variable calculated for images with no identifiable scapula because the other 3 variables (point picked, point 30 below, distance between coordinates) cannot be calculated if there is no identifiable scapula in a specific image.

For each of the variables assessed: pixel intensities of the total image, point picked (representative of the spine or medial border of the scapula), and the point 30 below the point picked, as well as the distance the scapula moved, the following parameters were calculated: mean, maximum, minimum, and standard deviation (**Table 1** and **Table 2**).

Table 1. Summary Data[1] for Pre-Processing of Manually Identifiable Images per Subject for each Testing Position

Testing Position: **Rest**					
Number of Subjects Processed	*Parameter*	Variables			
		Total Image Pixel Intensity	Point Picked Pixel Intensity	Point 30 Below Pixel Intensity	Distance between Coordinates of Identified Scapular Border (pixels)
16	*Mean ± Stdev*	49.55 ± 5.3	149.45 ± 43.4	64.85 ± 15.9	11.82 ± 19.53
	Min, Max	36.80, 60.33	54.81, 248.88	27.50, 127.81	0.31, 195.19
Testing Position: **Static Humeral Elevation in Scapular Plane**					
16	*Mean ± Stdev*	47.79 ± 5.1	128.47 ± 51.5	53.64 ± 20.1	14.69 ± 23.1
	Min, Max	37.25, 58.27	28.38, 246	16, 114.5	0.19, 241.24
Testing Position: **Dynamic Humeral Elevation in Scapular Plane**					
14	*Mean ± Stdev*	50.58 ± 7.2	159.70 ± 43.8	66.97 ± 18.1	14.23 ± 17.3
	Min, Max	30.99, 69.90	40.79, 254.21	15.71, 146.64	0, 308.44

Table 2. Summary Data[2] for Pre-Processing of all Images with No Identifiable Scapula per Subject for each Testing Position

Testing Position: **Rest**		
Number of Subjects Processed	*Parameter*	Variable
		Total Image Pixel Intensity
16	*Mean ± Stdev*	41.40 ± 8.8
	Max	55.17
Testing Position: **Static Humeral Elevation in Scapular Plane**		
16	*Mean ± Stdev*	44.09 ± 10.4
	Max	62.21
Testing Position: **Dynamic Humeral Elevation in Scapular Plane**		
14	*Mean ± Stdev*	46.35 ± 12.9
	Max	75.97

[1] Summary data was calculated using the averages from raw pre-processing data reported in Appendix E
[2] Summary data was calculated using the averages from raw pre-processing data reported in Appendix E

Analyses of these parameters were performed to determine how to optimize inclusion of the aforementioned variables. The parameters of the total image intensity variable were compared between images with identifiable scapular borders and those with no identifiable scapular border to determine an acceptable range and suitable cutoff value to help guide the automated program. Evaluation also occurred across subjects and style of imaging (static vs dynamic) as there were other factors to consider. For instance, some variables which are associated with pixel intensities may have a large variance due to the quality of ultrasound imaging. As reported in the previous manuscript[20], body composition can affect the quality of ultrasound images because significant adipose tissue or muscle mass can affect impedance. In visual analysis of our images, we observed that there was often times a bright region about 30 pixels above the scapular border which may present a likely source of error for the automated program. However, we noted that due to the hyperechoic appearance of bone in ultrasound images, there was always a significantly darker region about 30 pixels below the scapular border where bone was no longer visible. These observations led to evaluation of the parameters for the variable representing the pixel intensity of the point 30 pixels below the scapular border.

All variables and parameters were used, in conjunction with visual analysis of our ultrasound images, to help define appropriate pixel intensity cutoffs and regions of interest (ROIs). For the 3 variables associated with pixel intensity strength, evaluation of the pre-processing parameter values, comparison between the two types of images (identifiable and unidentifiable), evaluation of how many images were more (or less) than 1 standard deviation away from the mean, as well as trial and error were used to determine the pixel intensity cutoffs used in the automated program (**Table 3**). Through knowledge of how far apart the points picked to represent the scapular border were from frame to frame and that the scapular border was typically one of the brightest sections in our ultrasound images, we were able to combine displacement tracking and means of pixel intensities to define ROIs. We also created correction factors to weight the pixels in each ROI to eliminate automated selections which were likely to be erroneous.

Table 3. Pixel Intensity Cutoff Values Determined from Pre-Processing to Optimize Variables Included in the Automated Program

	Pixel Intensity Cut-off	
Variable	*Static*	*Dynamic*
Total Image	> 29	> 27
Point 30 Below	< 68	< 76
Picked Point	> 90	> 115

2.5.2 Automated Point Selection

Our novel automated point selection program was developed using results from our preliminary analyses of the characteristic features of our 2D shoulder ultrasound images and the location within these images that a manual (human) rater identified as the scapular border. Our proposed algorithm scans regions of interest based on the distance from the location of the last identified scapular border and weights clusters of averaged pixel intensities to estimate the location of the scapular border in the present image. We implemented a ‘chunking’ (clustering) method because performing an exhaustive scan of pixel intensities throughout a large ROI would be computationally expensive. Additionally, we developed a ‘correction factor’ method which was used to weight the pixel intensities in a given region of interest. Our correction factor method was based on our observation that there is a high likelihood that the scapular border in a subsequent image will be closer to its last located position from the previous image rather than farther away. The further away from the last located position, the greater the chance of error; for example, there is a higher possibility that the automated program will find and select a miscellaneous bright (white) region in the ultrasound image, the further it searches from the last selected location.

2.5.2.1 Chunking Implementation

We identified 4 primary regions of interest for the automated point selection program: an 81-by-81 pixel region, a 27-by-27 pixel region, a 9-by-9 pixel region, and a 3-by-3 pixel region. In the 81-by-81 pixel region, the central pixel is the location of the x-y coordinate chosen in the preceding image as the scapular border. Our chunking method begins by performing minimal scanning on our largest region of interest (ROI-81) in search of the ‘chunk’ with the greatest magnitude of mean pixel intensities, this 27-by-27 chunk (ROI-27) is then selected. This chunking and scanning process is then repeated for a selection of a 3rd, and smaller, region of interest (ROI-9). Within ROI-9, exhaustive scanning of each 3-by-3 pixel region of interest is performed. From within the final 3-by-3 pixel region chosen, the central pixel is chosen as the x-y coordinate representative of the scapular border.

2.5.2.2 Correction Factor Implementation

Prior to execution of the scans for ROIs 81, 27, and 9, the aforementioned correction factors were applied to each region to weight the pixel intensities. Our correction factor system is designed as nine 3x3 matrices with each cell in a matrix containing a value of either 1, 0.9, 0.8, 0.7, 0.6, or 0.5 (**Figure 2**). We will use a letter (A – I) to refer to each of the 9 correction factor matrices (**Figure 3**). Within each of these 3x3 matrices (A – I), the cell position will be referred to using indicial notation: 'A_{ij}', where 'i' represents the row (1-3) and 'j' represents the column (1-3) (**Figure 4**). Each ROI within an ultrasound image can be displayed as a matrix of pixel intensities. A matrix of pixel intensities that represents an ROI is multiplied by the appropriate correction factor matrix to generate a new matrix of weighted pixel intensities.

The underlying principle for the values, 0.5-1, that are in each cell of a correction factor matrix is as follows: In ROI-81, there are 9 27x27 regions among which the central pixel is the location of the last identified scapular border. In Matrix E, position E_{22}, the central cell contains the value 1 because this was the location of the last picked scapular border in the previous image so we estimate that the corresponding 27x27 region in ROI-81 has the highest likelihood of containing the location of the scapular border in the current image. The surrounding regions (E_{12}, E_{21}, E_{32}, and E_{23}) that have a distance of 1 from the cell that represents the location of the last picked scapular border will be multiplied by their corresponding value of 0.9. The further away from the most probable region, the lower the value (or weight) that a region is multiplied by. If the 27x27 region that corresponds to position E_{13} is still chosen, despite its lower weight, then each of the 9 9x9 regions in that 27x27 ROI is multiplied by matrix C. Similarly, if position E_{32} is chosen, then matrix H is applied to the next ROI.

0.5	0.6	0.7	0.6	0.5	0.6	0.7	0.6	0.5
0.6	0.8	0.9	0.7	0.8	0.7	0.9	0.8	0.6
0.7	0.9	1	0.9	1	0.9	1	0.9	0.7
0.5	0.7	0.9	**0.8**	**0.9**	**0.8**	0.9	0.7	0.5
0.6	0.8	1	**0.9**	**1**	**0.9**	1	0.8	0.6
0.5	0.7	0.9	**0.8**	**0.9**	**0.8**	0.9	0.7	0.5
0.7	0.9	1	0.9	1	0.9	1	0.9	0.7
0.6	0.8	0.9	0.7	0.8	0.7	0.9	0.8	0.6
0.5	0.6	0.7	0.6	0.5	0.6	0.7	0.6	0.5

Figure 2. Correction Factor Matrices

A	B	C
D	**E**	F
G	H	I

Figure 3. Labeled Matrices to Represent Correction Factor Matrix System

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

Figure 4. Inner Matrix Notation for Correction Factor Matrix System

2.5.2.3 Automated Algorithm

The custom automated point selection program began with an image being read into the program. For the first 5 frames, the identification of the scapular border must be performed by a manual rater. These first 5 frames require manual input to account for a possible syncing period in our ultrasound trials. If the 5th frame does not have an identifiable scapula, because the probe had not been placed back on the scapula by this frame or for any alternative reason, then manual input continues to be obtained until a frame with an identifiable scapula has been processed by the human rater. After an image with an identifiable scapula is processed, the automated program has a viable point of reference to start from. In the next image, the program assesses the image's total intensity to ensure that it is above the pre-defined total image intensity cutoff value (29 for static trials, 27 for dynamic trials). If the image is below the total cutoff range then the program records that there is no identifiable scapula in this image and proceeds to the next image. In the event that there was no identifiable scapula recorded in an image, the program obtains manual input for the next image (or however many images are needed to pass through until an identifiable image is encountered) to set it back on the right track. If the image was above the total cutoff range then program would proceed by creating ROI-81.

ROI-81 is the 81-by-81 pixel region around the location where the last identifiable scapula in the previous image was found. If at any time during the execution of the code, ROI-81 cannot be created because the last point picked is too close to the image borders, then manual input is obtained to put the program back on track. ROI-81's matrix of image intensities is clustered into 9 27x27 regions for which the mean pixel intensities are calculated. The appropriate correction factor matrix is then applied to ROI-81 by multiplying each of the averaged pixel intensity clusters by the corresponding correction factor to generate a weighted pixel intensity ROI-81 matrix. Each 27x27 region is then compared and the region with the greatest weighted average intensity is selected to be ROI-27. This method is then repeated for ROI-27 but for its 9 9x9 regions. If during either scanning process for ROI-81 or ROI-27 all 9 clusters are found to have the same mean pixel intensity value, the automated program would have recorded that there was no identifiable scapula for the particular image and proceeded to obtain manual input for the next image.

When the program has narrowed down one 9x9 region, ROI-9, a moving scan is performed, moving one pixel at a time, to cluster and average each 3x3 region that can be created within the ROI-9. When each 3x3 region is compared and a single 3x3 region is selected, the central pixel of this region is recorded as the estimated x-y coordinate representing the scapular border. The pixel intensity of the point 30 pixels below this x-y coordinate would then be evaluated to ensure that it is not greater than its pre-defined cutoff value (68 for static trials, 76 for dynamic). If the pixel intensity is greater than the cutoff value then manual input is obtained for the present image. If the point 30 below is not greater than the cutoff, then the final cutoff comparison is performed before confirming selection of the x-y coordinate. The pixel intensity of the x-y coordinate (point picked) is evaluated to confirm that it is above its pre-defined cutoff value (90 for static trials, 115 for dynamic). If this point picked is below the cutoff, then manual input is obtained. If this point picked is above the cutoff, then it is selected and recorded as the x-y coordinate representative of the scapular border in that image. For every image, the automated program records whether manual or automatic selection was used and each set of picked coordinates.

2.6 DATA ANALYSIS

2.6.1 Qualitative Analysis

Duplicated ultrasound images were generated for each subject and trial during execution of the automated point selection program. We plotted points representing the x-y coordinate that the manual rater identified as the scapular border in Worobey's study[20], the overall region of interest (ROI-81) that the automated program selected, and the x-y coordinate that the automated program identified as the scapular border on each ultrasound image (see example in **Figure 5**). These duplicated images were generated to assist with validation of the points picked by the automated program through visual analysis.

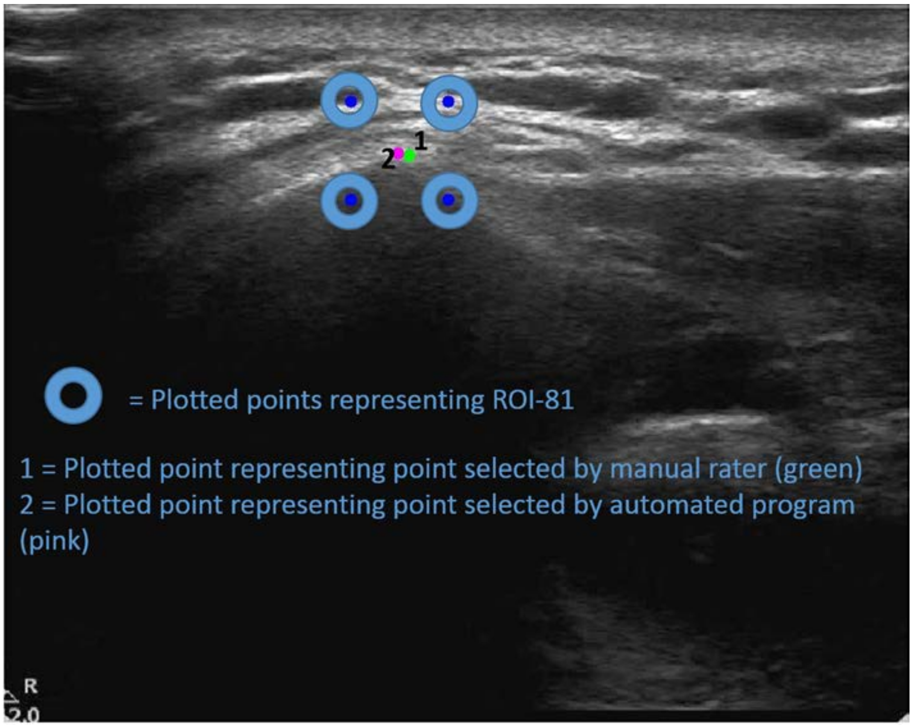

Figure 5. Example of Duplicated Ultrasound Image with ROI, Manual and Automated Point Selections of Scapular Border Plotted on Image

2.6.2 Quantitative Analysis

A custom Matlab program was written for post-processing (Appendix D) of all the data output from the automated point selection program. For all subjects processed, the following was calculated per trial: total number of images processed, amount of images with no identifiable scapula (as determined by the human rater), amount of images where manual input was needed to guide the automated program, amount of images where manual input was needed (not including include images that had no identifiable scapula), number of images for which the human rater selected as having no identifiable scapular border but the automated program reported there was an identifiable scapula (and vice versa), and the distances between the x-y coordinates selected by the manual rater versus the automated program. The elapsed time for the execution of the automated program was also measured using Matlab's stopwatch timer functions.

2.6.2.1 Statistical Analysis

Intraclass correlation coefficients (ICC) were calculated to evaluate the inter-rater reliability of the automated point selections as compared to the manual point selections. ICC(2,1) for single measures, based on the two-way random effects ANOVA model, was used to measure the extent of absolute agreement or interchangeability of the 2 methods of rating[32]. An ICC is measured on a scale of 0 to 1 where 0 indicates no reliability and 1 indicates perfect reliability with no measurement of error. To maintain consistency with the preceding study by Worobey[20], the ICC was interpreted using the categories for strength of agreement by Landis and Koch[33]. As reported by Landis and Koch, ICC values ≤ 0.4 are unacceptable, = 0.41-0.60 are moderately acceptable, = 0.61-0.80 are substantially acceptable, and ≥ 0.81 are almost perfect/excellent reliability. The distance between each x-y coordinate (selected by either the manual or automated rater) and the origin of the ultrasound image was calculated for ICC evaluations. Intraclass correlations were calculated for each testing position and participant trial, to evaluate all images which both methods of rating indicated as having an identifiable scapular border. Images that used manual input or assistance were not included in this analysis. The root mean square error (RMSE) was also calculated for all images with identifiable scapular borders per participant trial and testing position that did not use manual input. A cross-tabulation analysis with the Pearson Chi-Square test was also performed to assess the number of agreements and disagreements that occurred between the human and automated rating programs regarding the existence of an identifiable scapular border in an image. All statistical analyses were completed with a significance level set a priori at 0.5, and using IBM SPSS Statistics Software, version 21 (IBM Corporation; Armonk, New York).

3.0 RESULTS

Twenty-nine participants, 16 manual wheelchair users (MWUs) and 13 able-bodied (AB) individuals (gender = 13 females, 16 males; mean ± standard deviation age = 50.62 ± 10.5 years, height = 1.73 ± 0.1m, BMI = 23.65 ± 3.3), were included in the static trials for this study. Fourteen participants, 7 MWUs and 7 AB individuals (gender = 6 females, 8 males; mean ± standard deviation age = 53.36 ± 10.6 years, height = 1.73 ± 0.1m, BMI = 23.37 ± 2.6), were included in the dynamic trials for this study.

Static trials included 29 subjects, 1 trial per subject, for each of the 4 ultrasound scanning test positions: the participants arm in the resting position, elevated in the scapular, frontal, and sagittal planes. There was an average of 487 total images per static trial, about 25% of those images were recorded by a human rater as having no identifiable scapula. As a result, for static trials, about 50% of the overall trial required manual input, with approximately half of these images being images with no identifiable scapular border. The time required for data processing using the semi-automated detection program for a static trial was between 5 and 6 minutes.

Dynamic ultrasound scanning was performed during humeral elevation in the scapular plane. In the 14 dynamic trials (1 trial per subject), there was an average of 4,442 images per trial, about 21% of these images were documented as having no identifiable scapula by a human rater. Approximately 46% of the complete dynamic trials required manual input. Similar to the static trials, about half of these images were accounted for by images with no identifiable scapular border. The time required for data processing using the semi-automated detection program for a dynamic trial was roughly 50-90 minutes. The data used to perform these calculations can be found in extended tables in Appendix F; averages of these data values were used to create a summary data table, shown in **Table 4**.

Table 4. Summary Data for Post-Processing Results for each Testing Position

Averages	**Rest**	**Static Humeral Elevation in the Scapular Plane**	**Static Humeral Elevation in the Frontal Plane**	**Static Humeral Elevation in the Sagittal Plane**	**Dynamic Humeral Elevation in the Scapular Plane**
Total Number of Subjects Processed	29	29	29	29	14
Number of Images Processed per Subject	508	478	498	464	4,442
Number of Manually Identifiable Images	393	359	364	341	3,534
Number of Images Auto Identified	406	373	380	359	3,662
Number of Images Man = Yes ID, Auto = No ID	0	1	1	0	8
Number of Images Manually Unidentifiable Images	115	118	133	122	907
Number of Images Auto Unidentifiable Images	102	104	118	105	779
Number of Images Man = No ID, Auto = Yes ID	13	15	16	17	136
Number of Images Manual Input Requested	270	225	249	229	2,034
Number of Images Manual Input Used for Identifiable Image	169	121	133	125	1,265

Abbreviations:
Man =Manual/Human Rater; Auto = Automated Program
Yes ID = Identifiable Scapular Border Found; No ID = No Identifiable Scapular Border Found in Image

Intraclass correlations for inter-rater reliability between the manual and automated point selection methods were performed for each subject, in each testing position. These ICCs included all images where both raters (manual and auto) indicated that an identifiable scapular border existed. In the rest testing position, an average of 224 images were included per participant trial in the ICC analysis: all 29 participant trials demonstrated almost perfect reliability with ICCs ranging 0.811-0.995. For static humeral elevation in the scapular plane testing, an average of 238 images per participant trial were included: 5 of the 29 trials demonstrated substantial reliability with strengths ranging 0.629-0.760, the remaining 24 of 29 trials had excellent correlations ranging from 0.850 – 0.992. In the frontal plane, an average of 230 images per participant trial were included: 5 of the 29 trials were in the substantial reliability ICC category with strengths of 0.630 - 0.765, the remaining 24 of 29 ICCs were in the almost perfect range of 0.814 – 0.998. In the sagittal plane, an average of 216 images per participant trial were included: 6 of the 29 trials demonstrated moderate reliability with strengths between 0.635 – 0.743, the remaining 23 of 29 trials displayed excellent reliability ranging 0.810 – 0.992. Lastly, an average of 2,262 images per participant trial were included in the dynamic trials: all 14 dynamic trials revealed almost perfect reliability ranging 0.852 – 0.980. A summary of these ICCs split into trials is presented in **Table 5**. The number of images where both rating methods (manual & automated) indicated that an identifiable scapular border existed is for each participant trial, per testing position, is presented in **Table 6**.

Table 5. Intraclass Correlation Coefficients for Inter-Rater Reliability between Manual and Automated Point Selections of the Scapular Border across Testing Positions Split into Cases for each Subject

	Static Testing Positions				**Dynamic Motion in the Scapular Plane**
		Elevated Testing Positions			
Subject #	**Rest**	**Scapular Plane**	**Frontal Plane**	**Sagittal Plane**	
1	.811	.949	.981	.950	.956
2	.964	.947	.958	.955	.957
3	.960	.885	.977	.948	
4	.953	.968	*.675*	.924	
5	.975	.992	.998	.992	.980
6	.979	.959	.976	.983	.977
7	.982	.985	.985	*.692*	.980
8	.942	.915	.814	.819	
9	.980	.927	.835	.955	
10	.982	.908	.926	*.703*	.946
11	.986	.947	.972	.953	
12	.974	.960	.988	.933	.968
13	.972	.977	.954	*.635*	.945
14	.991	.986	.991	.990	
15	.990	*.737*	*.630*	.978	
16	.983	*.723*	*.759*	.810	
17	.977	.974	.957	.957	
18	.980	.939	.869	.963	.952
19	.992	.985	.979	.938	.963
20	.983	*.760*	.968	.962	
21	.991	.920	.988	.920	
22	.995	.975	.990	.982	.975
23	.945	.955	.981	.945	
24	.988	.986	.875	.974	.903
25	.866	*.629*	*.697*	*.690*	.870
26	.942	.861	.893	*.644*	
27	.988	*.635*	*.765*	*.743*	
28	.941	.850	.876	.943	
29	.985	.913	.940	.934	.852

Table 6. Total Number of Images with Identifiable Scapular Border as Indicated and Agreed Upon by both Methods of Point Selection

	Static Testing Positions				**Dynamic Motion in the Scapular Plane**
		Elevated Testing Positions			
Subject #	**Rest**	**Scapular Plane**	**Frontal Plane**	**Sagittal Plane**	
1	142	213	197	233	1547
2	295	175	215	203	1682
3	102	122	127	92	
4	196	156	134	115	
5	105	98	126	151	2283
6	105	92	113	98	2178
7	143	123	131	89	2419
8	121	140	110	135	
9	166	147	128	102	
10	140	155	127	131	2719
11	184	190	241	159	
12	141	247	220	252	2208
13	241	323	194	211	3125
14	259	246	156	102	
15	100	211	179	186	
16	263	232	333	237	
17	262	254	284	293	
18	160	286	242	262	2437
19	245	214	295	275	2024
20	333	416	271	236	
21	334	332	288	381	
22	218	284	270	260	1993
23	291	270	224	150	
24	283	187	404	337	2352
25	472	413	320	344	2468
26	299	351	337	338	
27	261	316	411	294	
28	271	334	234	262	
29	350	362	373	347	2233

The distance between each point (x-y coordinated) chosen to represent the scapular border by the automated versus human rater was calculated for every identifiable image in each participant trial per testing position. The total mean (± stdev) distance between the x-y coordinate that the manual rater versus the x-y coordinate that the automated rater chose was 25.6 ± 31.9 pixels for static trials. The scaling factor used in Worobey's study was about 14.5 pixels per millimeter. Thus, this mean distance is about 1.77mm. For dynamic trials, the total mean distance was 22.2 ± 28.1 pixels; accordingly, the mean distance was about 1.53mm. The mean distance for each testing position is shown in **Table 7**.

Table 7. Distance between Points Picked by Human vs. Automated Point Selection Methods (Mean ± Standard Deviation) for all Participant Trials across Testing Positions

Testing Position	Mean ± stdev (pixels)	Mean ± stdev (mm)
Rest	17.4 ± 8.9	1.2 ± 0.6
Static Scapular Humeral Elevation	29.3 ± 37.9	2.0 ± 2.6
Static Frontal Humeral Elevation	27.2 ± 36.5	1.9 ± 2.5
Static Sagittal Humeral Elevation	28.3 ± 44.4	2.0 ± 3.1
Dynamic Humeral Elevation	22.2 ± 28.1	1.5 ± 1.9

The root mean square error (RMSE) was also calculated for each participant trial across testing positions. The average RMSE for each testing position was 16.43 pixels (1.13mm) in the rest testing position, 31.19 pixels (2.15mm) in static scapular elevation, 29.91 pixels (2.06mm) in static frontal elevation, 32.93 pixels (2.27mm) in static sagittal elevation, and 27.11 pixels (1.87mm) in dynamic scapular elevation. The complete table of these RMSEs can be found in **Table 8**.

Table 8. Root Mean Square Errors (in pixels) between Manual and Automated Point Selections of the Scapular Border across Testing Positions Split into Cases for each Subject

	Static Testing Positions				**Dynamic Motion in the Scapular Plane**
		Elevated Testing Positions			
Subject #	**Rest**	**Scapular Plane**	**Frontal Plane**	**Sagittal Plane**	
1	37.0	22.1	19.8	28.7	28.9
2	27.0	34.6	28.3	26.6	22.8
3	15.5	50.0	20.8	34.0	
4	22.2	29.1	89.7	24.5	
5	11.5	12.1	9.1	10.9	18.8
6	13.4	24.1	17.5	15.6	17.1
7	13.2	22.2	15.0	54.6	20.2
8	14.8	28.3	48.8	34.7	
9	17.3	46.2	59.7	33.2	
10	15.5	36.0	24.2	63.7	23.0
11	11.6	28.2	21.2	14.0	
12	15.7	15.3	10.4	24.5	16.7
13	27.1	26.0	30.3	98.1	37.5
14	12.3	15.1	14.1	10.4	
15	9.4	40.8	47.6	16.1	
16	11.6	59.3	31.5	35.5	
17	14.0	16.7	20.4	19.4	
18	13.6	22.2	25.7	14.8	32.1
19	13.4	15.7	17.8	29.5	21.4
20	11.7	39.0	20.8	18.9	
21	11.2	30.3	21.3	31.0	
22	7.7	17.9	12.4	19.3	19.1
23	16.0	20.2	14.7	14.2	
24	10.5	16.3	40.6	21.8	40.4
25	34.3	51.2	61.4	51.3	33.5
26	20.9	46.0	27.4	92.6	
27	10.6	67.8	51.8	74.0	
28	23.3	36.3	38.5	20.3	
29	14.1	35.5	26.7	22.7	48.0

A cross-tabulation analysis was conducted for each of the 5 testing positions. These analyses included all images processed per testing position, images with and without identifiable scapular borders. This overall analysis was to evaluate how often the two methods of point selection (human vs. automated) agreed and disagreed on the existence of an identifiable scapular border in each processed image. In the 4 static testing positions, a total of 56,466 images were processed (about 14,116 per testing position); within these images, about 87.4% of the images where the manual rater said there was no identifiable scapular border in an image, the automated program agreed. However, about 12.6% of these images that the human rater indicated as having no identifiable scapular border, the automated program indicated that an identifiable border existed. Within the images where the human rater indicated that an identifiable scapular border existed, the automated program agreed about 99.8% of the time. In the dynamic testing position, a total of 62,184 images were processed; within these images, there was an 85% agreement between the images that the human vs automated rater reported as missing an identifiable scapular border. For the images that the human rater reported as having an identifiable scapular border, the automated program agreed 99.8%. The Pearson Chi-Square p-value for these 5 cross-tabulation analyses was 0.000. The results for the cross-tabulation analysis, for each testing position, are displayed in **Table 9 - Table 13**.

Table 9. Cross-Tabulation Analysis Results for Existence of Identifiable Scapular Border as Determined by Human versus Automated Rater for All Images in the Rest Testing Position

Static Trials for the Rest Testing Position = Total images for 29 subjects			**autoID** = As determined by automated program		Total
			0 = No Identifiable Scapular Border in Image	1 = Identifiable Scapular Border in Image	
manID = As determined by human rater	0 = No Identifiable Scapular Border in Image	Count	2937	387	3324
		% within manID	88.4%	11.6%	100.0%
	1 = Identifiable Scapular Border in Image	Count	10	11392	11402
		% within manID	0.1%	99.9%	100.0%
Total		Count	2947	11779	14726
		% within manID	20.0%	80.0%	100.0%

Table 10. Cross-Tabulation Analysis Results for Existence of Identifiable Scapular Border as Determined by Human versus Automated Rater for All Images in the Static Humeral Elevation Scapular Plane Testing Position

Static Trials for Humeral Elevation in Scapular Plane = Total images for 29 subjects			**autoID** = As determined by automated program		Total
			0 = No Identifiable Scapular Border in Image	1 = Identifiable Scapular Border in Image	
manID = As determined by human rater	0 = No Identifiable Scapular Border in Image	Count	3006	426	3432
		% within manID	87.6%	12.4%	100.0%
	1 = Identifiable Scapular Border in Image	Count	19	10404	10423
		% within manID	0.2%	99.8%	100.0%
Total		Count	3025	10830	13855
		% within manID	21.8%	78.2%	100.0%

Table 11. Cross-Tabulation Analysis Results for Existence of Identifiable Scapular Border as Determined by Human versus Automated Rater for All Images in the Static Humeral Elevation Frontal Plane Testing Position

Static Trials for Humeral Elevation in Frontal Plane = Total images for 29 subjects			**autoID** = As determined by automated program		Total
			0 = No Identifiable Scapular Border in Image	1 = Identifiable Scapular Border in Image	
manID = As determined by human rater	0 = No Identifiable Scapular Border in Image	Count	3392	477	3869
		% within manID	87.7%	12.3%	100.0%
	1 = Identifiable Scapular Border in Image	Count	31	10536	10567
		% within manID	0.3%	99.7%	100.0%
Total		Count	3423	11013	14436
		% within manID	23.7%	76.3%	100.0%

Table 12. Cross-Tabulation Analysis Results for Existence of Identifiable Scapular Border as Determined by Human versus Automated Rater for All Images in the Static Humeral Elevation Sagittal Plane Testing Position

Static Trials for Humeral Elevation in Sagittal Plane = Total images for 29 subjects			**autoID** = As determined by automated program		Total
			0 = No Identifiable Scapular Border in Image	1 = Identifiable Scapular Border in Image	
manID = As determined by human rater	0 = No Identifiable Scapular Border in Image	Count	3043	504	3547
		% within manID	85.8%	14.2%	100.0%
	1 = Identifiable Scapular Border in Image	Count	5	9897	9902
		% within manID	0.1%	99.9%	100.0%
Total		Count	3048	10401	13449
		% within manID	22.7%	77.3%	100.0%

Table 13. Cross-Tabulation Analysis Results for Existence of Identifiable Scapular Border as Determined by Human versus Automated Rater for All Images in the Dynamic Humeral Elevation Scapular Plane Testing Position

Dynamic Trials = Total images for 14 subjects			**autoID** = As determined by automated program		Total
			0 = No Identifiable Scapular Border in Image	1 = Identifiable Scapular Border in Image	
manID = As determined by human rater	0 = No Identifiable Scapular Border in Image	Count	10803	1900	12703
		% within manID	85.0%	15.0%	100.0%
	1 = Identifiable Scapular Border in Image	Count	107	49374	49481
		% within manID	0.2%	99.8%	100.0%
Total		Count	10910	51274	62184
		% within manID	17.5%	82.5%	100.0%

4.0 DISCUSSION

The overall objective for this study was to create an automated (or semi-automated) alternative to the manual point selection method in order to increase the efficiency of the data processing of our ultrasound images for the Freehand Three-Dimensional Ultrasound (3D FUS) System. This alternative point selection method was to be created with a balance between accuracy and efficiency as compared to its gold standard (manual). In addition, the reliability of this automated point selection method needed to be evaluated. While using the automated point selection method increases the efficiency of data processing by requiring as much as 63-79% less time than the manual point selection method, the inter-rater reliability strength decreases. The inter-rater reliability strength between multiple human raters using the manual point selection method is reported in Worobey's study as almost perfect, ranging 0.975-0.995, whereas the ICC calculated in this study between the automated point selection program and a human rater ranges 0.629-0.998 which is within the substantial to almost perfect reliability category. Overall, our ICC and cross-tabulation results indicate that the automated point selection method displays substantial to excellent inter-rater reliability as well as about 85-100% agreement with the manual point selection gold standard.

During our ultrasound imaging procedure, there were multiple instances when the ultrasound probe was not in contact with the scapula, mainly due to rotation of the probe from the spine of the scapula to the medial border. These instances result in a variable number of frames where there is no identifiable scapula. The number of frames in an ultrasound video or sequence of images that has no identifiable scapula directly affects the amount of manual input that is required throughout the execution of the automated program. As reported in Chapter 3 of this study, images with no identifiable scapula account for about 25% of an ultrasound image set, these images have the potential to approximately double the amount (25% to 50%) of manual input that is required during the automated program. Fortunately, images with no identifiable scapula typically occur as a large number of consecutive frames; therefore, while the number of selections required of a human rater have doubled, the actual time spent by and required of the human rater may not double.

In Chapter 3, in our cross-tabulation analysis, we also report that for about 13-15% of the images that a manual rater has indicated as having no identifiable scapula, the automated program disagreed. While some of these images did in fact have no identifiable scapula, we found through our qualitative analysis that some did have an identifiable scapula. The majority of these images with this discrepancy between the two methods of point selection are of a lower quality. In a low quality ultrasound image it is increasingly difficult to pinpoint the exact location of the scapular border. There are numerous factors that may negatively affect the quality of an ultrasound image but one that we identified is body composition. Another possible explanation for the discrepancy may be human error. We observed that in some of these images the automated program would select an x-y coordinate as the scapular border and in the subsequent image the manual rater had selected approximately the same location. It is possible that as the manual rater was processing any set of ultrasound images, he or she overlooked a few images that in actuality had an identifiable scapular border.

During evaluation of the qualitative analysis results, it was also noted that the automated program did not perform as well when the medial border of the scapula was being scanned versus when the lateral (spine) border of the scapula was being imaged. The corresponding images where the automated program had a lower performance also typically had many more bright regions that were close together (or closer to the last picked point) and therefore served as additional sources of error for the automated program.

Data processing, in regards to identifying the scapular border in our ultrasound images, was a very time consuming process when the procedure was entirely manual. The manual method required roughly 30 minutes for processing a static trial and 240 minutes (4 hours) to process a dynamic trial. The semi-automated method minimizes the amount of time required for this phase of data processing by about 63-79%, requiring roughly 5-6 minutes for a static trial and 50-90 minutes for a dynamic trial. The time required for processing using our proposed semi-automated algorithm may vary due to the amount of manual input required in a specific trial. Additionally, we measured the elapsed time for execution of the semi-automated program using Matlab's stopwatch timer functions which may be adversely affected if there are other applications running in the background of your system. The time required for the semi-automated program to run may also be affected by the general speed of the computer system. It is also important to note that for the purpose of this study, the semi-automated program included the process of duplicating each ultrasound image in a trial for our qualitative analysis, as well as calculating and reporting the distances between both methods (auto and manual) of points picked. As these components of the program would not be included in the semi-automated program that would be run in a normal data processing routine, the time required for processing using the automated program may decrease. Conversely, the version of the semi-automated program used for this study was not entirely interactive as it used pre-selected manual points when manual input was requested in order to be consistent with the points selected in the previous study by Dr. Worobey which served as the gold standard for this study. During a normal run of the semi-automated program that will require new manual input during data processing, the time required for the program to run will likely increase. The version of the automated point selection program to be used in a normal data processing scenario is included in Appendix G. Given the aforementioned variations of time required for the automated point selection method, the minimum and maximum amount of time required will increase from the average elapsed time reported.

The results of this study may be limited by a relatively small sample size. Including more participants/trials may provide a more robust analysis of the accuracy of our semi-automated algorithm. In addition, there were limiting factors on the ICC analysis conducted in this study. ICC values are likely higher because the images were not independent of each other. There was one ultrasound imaging video per subject but each video was written to multiple frames; therefore, multiple images from a single subject were included in the analysis which violates the statistical rule of independence. An alternative method of statistical analysis should be considered in future analysis to control for the association of images within a single subject. In addition, intra-rater reliability across human raters was not taken into account as pre-selected points from Worobey's study consisted of points from 3 different human raters.

Future application and enhancement of the proposed automated point selection method should include supplementary analysis to further establish and increase the program's accuracy. For supplementary qualitative analysis of our automated program, the process used in Worobey, 2014[20] should be applied to create a reconstructed 3D scapular border and determine scapular rotations to visually compare how the point selections between the two methods (manual vs auto) affect the reconstructed border and scapular rotations. Additionally, an analysis of duplicated images and participant demographics should be conducted to discover patterns of error. For example, an analysis of the correlation between a participant's BMI and the accuracy of the program during each trial corresponding to that patient should be performed. Further analysis on optimization of the pixel intensity cutoff values determined in the pre-processing phase of this study should be explored. Evaluation of the cutoff values may be performed to determine a method to personalize or customize the values per subject to increase their effectiveness. Inclusion of more intricate image pre-processing techniques may also enhance the accuracy of our automated program. Future study should explore techniques such as filtering[34], bone boundary extraction[35], speckle/noise removal[21], as well as edge, boundary, or corner detection algorithms[36], to be incorporated into our automated program. Continued development of the automated point selection program should also include alternative techniques to improve the program's balance between accuracy and efficiency. To decrease the amount of manual input required by the proposed method, an alternative ROI can be created and called when the point 30 below the estimated x-y coordinate is too bright or when ROI81 cannot be created due to its proximity to the image borders. A stop function may also be implemented to decrease the amount of images with unidentifiable scapular borders that manual input is required for or to increase the amount of manual assistance while increasing the accuracy of the program as a human rater observes the program veering off track (selecting points away from the scapular border). Future studies may also include broadening or slightly modifying our automated algorithm to make it applicable to ultrasound images that are unrelated to our 3D FUS custom system or further, to tracking other muscle or bony landmarks in 2D ultrasound images.

In conclusion, the automated point selection program proposed in this study provides a time-efficient means for identifying the location of the scapular border in 2D ultrasound images. Through our data analysis, we have demonstrated that our semi-automated program provides a balance between accuracy and efficiency. We have also established that our automated algorithm is a reliable method as compared to our gold standard of manual point selections. Intraclass correlations, with substantial to almost perfect reliability, validate that the previous manual point selection method can be replaced by our automated point selection program without significantly sacrificing accuracy.

Minimizing the time required for data processing was needed to enhance the data processing of our custom 3D FUS system. As rehabilitation professionals are still searching for a reliable, non-invasive method of scapular examination techniques[11], there is a clinical need for such a system. Our 3D FUS system has great potential for serving as a non-invasive means of evaluating static and dynamic scapular motion. Further development of our 3D FUS system may help evaluate the efficacy of interventions used to correct altered scapular kinematics, reduce injury risk, and treat shoulder pathology. As such, this tool could be of great value to clinicians.

APPENDIX A

MATLAB CODE FOR PRE-PROCESSING

```
clear
clc

TYPE = 3;  %1 is rest2, 2 is scap2, 3 is dynscap, 4 is front2, 5 is sag2

infofile = 'dynscap_info.xlsx';
info = xlsread(infofile);

% iRange = ['A',num2str(i),':G',num2str(i)];
for a = 15:29
    valid = isnan(info(a,2));
    if valid == 1
        disp(['Did not process count ', int2str(a)])
        continue
    end
    SUBJ = info(a,1)
    COUNT = info(a,7);
    framestart = info(a,2);
    framestop = info(a,3);
    separate = info(a,6);
    pointsTitle = info(a,5);
    imageTitle = info(a,4);

% SUBJ = 17;
% COUNT = 11; %what row to save
% framestart = 1;
% framestop = 300;
% namefile='spine_points_ml.xls'; %%where manual points are
% imagetitle = 'syncframe';
% separate = 1; %1 for yes, 2 for no

aPPfolder = 'Z:\Protocols Boninger\Ultrasound and dynamic shoulder movement
(262)\Data (Analysis)\March Transfer\pre proc';
x1Range = ['A',num2str(COUNT)];  %for writing in aPP folder
```

```
if imageTitle == 1
    imagetitle = 'syncframe';
elseif imageTitle == 2
    imagetitle = 'frame';
end

if pointsTitle == 1
    namefile = 'spine_points.xls';
elseif pointsTitle == 2
    namefile = 'spine_points_ml.xls';
elseif pointsTitle == 3
    namefile = 'spine_points_LS.xls';
end


if TYPE == 1
    PLANE = 'rest2';
    filenamePINT = 'REST2pint_h.xlsx';
    filenameNOID = 'REST2noid_h.xlsx';
    filenameDIST = 'REST2dist_h.xlsx';
    filenameWRITE = ['PREPROCESS_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];
end
if TYPE == 2
    PLANE = 'scap2';
    filenamePINT = 'SCAP2pint_h.xlsx';
    filenameNOID = 'SCAP2noid_h.xlsx';
    filenameDIST = 'SCAP2dist_h.xlsx';
    filenameWRITE = ['PREPROCESS_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];
end
if TYPE == 3
    PLANE = 'dynscap';
    filenamePINT = 'DYNSCAPpint_h.xlsx';
    filenameNOID = 'DYNSCAPnoid_h.xlsx';
    filenameDIST = 'DYNSCAPdist_h.xlsx';
    filenameWRITE = ['PREPROCESS_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];
end

subjfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic shoulder movement
(262)\Data (Investigator)\US Images\SCAP',num2str(SUBJ),'\',PLANE];
cd(subjfolder)
pickedpts=xlsread(namefile);

if separate == 1
    imgfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic shoulder
movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE,'\sync_frames'];
end
if separate == 2  %not separated
    imgfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic shoulder
movement (262)\ Data (Investigator)\US Images\SCAP',num2str(SUBJ),'\',PLANE];
end
cd(imgfolder)
```

```
%%%set by us; can be changed%%%
X1 = 227.5;  Y1 = 172.5;  X2 = 966.5;  Y2 = 760.5;  %standard values for
cropping
conversion=double(145.5); %convert excel value from cm to pix by
cm*conversion
box81 = (81/2) - 0.5;  %using 81x81 box as start so 40 above&below
box27 = (27/2) - 0.5;  %using 27x27 box as start so 13 above&below
box9 = (9/2) - 0.5;  %using 9x9 box as start so 4 above&below
EOI_x = (X2-X1)+1; EOI_y = (Y2-Y1)+1;  %end of image

% pint = zeros(framestop,7);
% noid = zeros(framestop,2);
pint = [];
noid = [];
c = 1;  %what row to save noid values
d = 1;  %what row to save identifiable values




framend = (framestop - framestart) + 1;
% frame = framestart;
imgFrame = framestart - 1;
for i = 1:framend
    frame = i + imgFrame;
    manual_xCM = double(pickedpts(i,1));
    manual_yCM = double(pickedpts(i,2));
    manual_x = double(pickedpts(i,1)*conversion);  %convert to pix
    manual_y = double(pickedpts(i,2)*conversion);
    manual_xy = manual_xCM + manual_yCM;

    %if no identifiable
    if manual_xy < 1
        img_orig = imread([imagetitle,num2str(frame),'.jpeg']); %read in
image
        img2gray = rgb2gray(img_orig);  %convert to grayscale
        img = img2gray(Y1:Y2,X1:X2);  %crop image using standard values
        ROI_orig = img;  %create matrix of image intensities
        noid(c,1) = double(mean(mean(ROI_orig)));  %total image average
        noid(c,2) = manual_xCM + manual_yCM;
        noid(c,3) = frame;
        c = c + 1;
%         disp(['Did not process subject ', int2str(SUBJ), ', no
identifiable'])
%         frame = frame + 1;
        %exit if & for loop and proceed to next i
        continue
    end

    %if identifiable
    if manual_xy > 1
        img_orig = imread([imagetitle,num2str(frame),'.jpeg']); %read in
image
        img2gray = rgb2gray(img_orig);  %convert to grayscale
        img = img2gray(Y1:Y2,X1:X2);  %crop image using standard values
```

```
        % imshow(img); hold on;
        % close()
        ROI_orig = img;  %create matrix of image intensities
        pint(d,1) = double(mean(mean(ROI_orig)));  %total image intensity

        pint(d,2) = double(ROI_orig(round(manual_y),round(manual_x)));
%manPicked point pix intensity

        below30y = manual_y + 30;
        pint(d,3) = double(ROI_orig(round(below30y),round(manual_x))); %pix
intensity 30 below point picked

        ROI81_X1 = manual_x - box81;  ROI81_X2 = manual_x + box81;
        ROI81_Y1 = manual_y - box81;  ROI81_Y2 = manual_y + box81;
        if ROI81_X1 > 0 && ROI81_Y1 > 0 && ROI81_X2 < EOI_x && ROI81_Y2 <
EOI_y
            img81 = img(ROI81_Y1:ROI81_Y2,ROI81_X1:ROI81_X2);
            pint(d,4) = double(mean(mean(img81))); %intensity of ROI81
        else
            pint(d,4) = NaN; %couldn't make ROI81 so put NaN

        end

        ROI27_X1 = manual_x - box27;  ROI27_X2 = manual_x + box27;
        ROI27_Y1 = manual_y - box27;  ROI27_Y2 = manual_y + box27;
        if ROI27_X1 > 0 && ROI27_Y1 > 0 && ROI27_X2 < EOI_x && ROI27_Y2 <
EOI_y
            img27 = img(ROI27_Y1:ROI27_Y2,ROI27_X1:ROI27_X2);
            pint(d,5) = double(mean(mean(img27))); %intensity of ROI27
        else
            pint(d,5) = NaN; %couldn't make ROI27 so put NaN
        end

        ROI9_X1 = manual_x - box9;  ROI9_X2 = manual_x + box9;
        ROI9_Y1 = manual_y - box9;  ROI9_Y2 = manual_y + box9;
        if ROI9_X1 > 0 && ROI9_Y1 > 0 && ROI9_X2 < EOI_x && ROI9_Y2 < EOI_y
            img9 = img(ROI9_Y1:ROI9_Y2,ROI9_X1:ROI9_X2);
            pint(d,6) = double(mean(mean(img9))); %intensity of ROI9
        else
            pint(d,6) = NaN; %couldn't make ROI9 so put NaN

        end

        pint(d,7) = manual_xCM + manual_yCM;
        pint(d,8) = frame;
        d = d + 1;
%         frame = frame + 1;
    end


end
```

```
pintLENGTH = length(pint);
manID = pint(:,8);
IDlength = pintLENGTH - 1;
% distances = zeros(IDlength);
distances = [];
zTemp = [];

for j = 1:IDlength
    k = j + 1;
    j1 = manID(j,1);  %what frame# is in the 1st row of the manually
identified column vector
    k1 = manID(k,1);
    x1 = pickedpts((j1-imgFrame),1)*conversion; y1 = pickedpts((j1-
imgFrame),2)*conversion;
    x2 = pickedpts((k1-imgFrame),1)*conversion; y2 = pickedpts((k1-
imgFrame),2)*conversion;
    distances(j,1) = sqrt((x2-x1).^2 + (y2-y1).^2); %distance scap moves
frame to frame
    distances(j,2) = abs(x2-x1); %how much did scap move in x frome frame to
frame
    distances(j,3) = abs(y2-y1); %how much did scap move in y frome frame to
frame
    distances(j,4) = j1;
    distances(j,5) = k1;
end


cd(subjfolder)
xlswrite(filenameWRITE,pint,'pint')
xlswrite(filenameWRITE,noid,'noid')
xlswrite(filenameWRITE,distances,'dist')

%calculates zs for dist
distAVG = mean(distances(:,1:3)); distAVG(1,4) = SUBJ; distAVG(1,5) =
pintLENGTH;
distSTD = std(distances(:,1:3)); distSTD(1,4) = SUBJ; distSTD(1,5) =
pintLENGTH;
distLENGTH = length(distances);
for m = 1:distLENGTH
    distZtemp(m,1) = (distances(m,1) - distAVG(1,1))/distSTD(1,1); %calculate
zscores
    distZtemp(m,2) = (distances(m,2) - distAVG(1,2))/distSTD(1,2);
    distZtemp(m,3) = (distances(m,3) - distAVG(1,3))/distSTD(1,3);
end
zsT = 0; zsX = 0; zsY = 0; %number of times zscore is > 1 (i.e. more than 1
away from std)
for n = 1:distLENGTH
    if distZtemp(n,1) > 1
        zsT = zsT + 1;
    end
    if distZtemp(n,2) > 1
        zsX = zsX + 1;
    end
    if distZtemp(n,3) > 1
```

```
        zsY = zsY + 1;
    end
end
distZ = [zsT zsX zsY SUBJ distLENGTH];
%calculate zs for pint
pintAVG = mean(pint(:,1:7)); pintAVG(1,8) = SUBJ; pintAVG(1,9) = pintLENGTH;
pintSTD = std(pint(:,1:7)); pintSTD(1,8) = SUBJ; pintSTD(1,9) = pintLENGTH;
for m = 1:pintLENGTH
    pintZtemp(m,1) = (pint(m,1) - pintAVG(1,1))/pintSTD(1,1); %calculate
zscores
    pintZtemp(m,2) = (pint(m,2) - pintAVG(1,2))/pintSTD(1,2);
    pintZtemp(m,3) = (pint(m,3) - pintAVG(1,3))/pintSTD(1,3);
end
zsTp = 0; zspp = 0; zs30p = 0; %number of times zscore is... (i.e. more than
1 away from std)
for n = 1:pintLENGTH
    if pintZtemp(n,1) < -1
        zsTp = zsTp + 1;
    end
    if pintZtemp(n,2) < -1
        zspp = zspp + 1;
    end
    if pintZtemp(n,3) > 1
        zs30p = zs30p + 1;
    end
end
pintZ = [zsTp zspp zs30p SUBJ pintLENGTH];
%calculate zs for noid
noidLENGTH = length(noid);
noidAVG = mean(noid(:,1:2)); noidAVG(1,3) = SUBJ; noidAVG(1,4) = noidLENGTH;
noidSTD = std(noid(:,1:2)); noidSTD(1,3) = SUBJ; noidSTD(1,4) = noidLENGTH;
for m = 1:noidLENGTH
    noidZtemp(m,1) = (noid(m,1) - noidAVG(1,1))/noidSTD(1,1); %calculate
zscores
end
zsTn = 0;  %number of times zscore is... (i.e. more than 1 away from std)
for n = 1:noidLENGTH
    if noidZtemp(n,1) > 1
        zsTn = zsTn + 1;
    end
end
noidZ = [zsTn SUBJ noidLENGTH];



cd(subjfolder)
xlswrite(filenameWRITE,pintZtemp,'Zpint')
xlswrite(filenameWRITE,noidZtemp,'Znoid')
xlswrite(filenameWRITE,distZtemp,'ZDist')


pintMAX = max(pint(:,1:7)); pintMAX(1,8) = SUBJ; pintMAX(1,9) = pintLENGTH;
pintMIN = min(pint(:,1:7)); pintMIN(1,8) = SUBJ; pintMIN(1,9) = pintLENGTH;
pintVAR = var(pint(:,1:7)); pintVAR(1,8) = SUBJ; pintVAR(1,9) = pintLENGTH;
```

```
noidMAX = max(noid(:,1:2)); noidMAX(1,3) = SUBJ; noidMAX(1,4) = noidLENGTH;
noidMIN = min(noid(:,1:2)); noidMIN(1,3) = SUBJ; noidMIN(1,4) = noidLENGTH;
noidVAR = var(noid(:,1:2)); noidVAR(1,3) = SUBJ; noidVAR(1,4) = noidLENGTH;

distMAX = max(distances(:,1:3)); distMAX(1,4) = SUBJ; distMAX(1,5) =
pintLENGTH;
distMIN = min(distances(:,1:3)); distMIN(1,4) = SUBJ; distMIN(1,5) =
pintLENGTH;
distVAR = var(distances(:,1:3)); distVAR(1,4) = SUBJ; distVAR(1,5) =
pintLENGTH;


cd(aPPfolder)
xlswrite(filenamePINT,pintAVG,'avgs',x1Range)
xlswrite(filenamePINT,pintMAX,'maxs',x1Range)
xlswrite(filenamePINT,pintMIN,'mins',x1Range)
xlswrite(filenamePINT,pintSTD,'stdevs',x1Range)
xlswrite(filenamePINT,pintVAR,'vars',x1Range)
xlswrite(filenamePINT,pintZ,'zs',x1Range)
xlswrite(filenameNOID,noidAVG,'avgs',x1Range)
xlswrite(filenameNOID,noidMAX,'maxs',x1Range)
xlswrite(filenameNOID,noidMIN,'mins',x1Range)
xlswrite(filenameNOID,noidSTD,'stdevs',x1Range)
xlswrite(filenameNOID,noidVAR,'vars',x1Range)
xlswrite(filenameNOID,noidZ,'zs',x1Range)
xlswrite(filenameDIST,distAVG,'avgs',x1Range)
xlswrite(filenameDIST,distMAX,'maxs',x1Range)
xlswrite(filenameDIST,distMIN,'mins',x1Range)
xlswrite(filenameDIST,distSTD,'stdevs',x1Range)
xlswrite(filenameDIST,distVAR,'vars',x1Range)
xlswrite(filenameDIST,distZ,'zs',x1Range)

disp(['Processed subject ', int2str(SUBJ)])



end
```

APPENDIX B

MATLAB CODE FOR AUTOMATED PROGRAM FOR A STATIC TRIAL

```
function CFchunk6

warning('off','all')
warning

clear
clc


%folder with autocode and ROI n fxn code n info
mainfolder = 'Z:\Protocols Boninger\Ultrasound and dynamic shoulder movement
(262)\Data (Analysis)\March Transfer\programs';

% TYPE = 1;  %1 is rest2, 2 is scap2, 5 is dynscap, 3 is front2, 4 is sag2
tic
for TYPE = 3
    disp(['TYPE ',num2str(TYPE)])
    cd(mainfolder)

if TYPE == 1
    PLANE = 'rest2';
    infofile = 'rest2_info.xlsx';
end
if TYPE == 2
    PLANE = 'scap2';
    infofile = 'scap2_info.xlsx';
end
if TYPE == 5
    PLANE = 'dynscap';
    infofile = 'dynscap_info.xlsx';
end
if TYPE == 3
    PLANE = 'front2';
    infofile = 'front2_info.xlsx';
```

```
end
if TYPE == 4
    PLANE = 'sag2';
    infofile = 'sag2_info.xlsx';
end


info = xlsread(infofile);
for a = 1:29
%     tic
    SUBJ = info(a,1)
    valid = isnan(info(a,2));
    if valid == 1
        disp(['Did not process subject ', int2str(SUBJ)])
        continue
    end
    framestart = info(a,2);
    framestop = info(a,3);
    imageTitle = info(a,4);
    pointsTitle = info(a,5);
    separate = info(a,6);
    COUNT = info(a,7);

    if imageTitle == 1
        imagetitle = 'syncframe';
    elseif imageTitle == 2
        imagetitle = 'frame';
    end

    if pointsTitle == 1
        namefile = 'spine_points.xls';
    elseif pointsTitle == 2
        namefile = 'spine_points_ml.xls';
    elseif pointsTitle == 3
        namefile = 'spine_points_LS.xls';
    end

    if TYPE == 1 || TYPE == 2
            subjfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic
shoulder movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE]; %store files
    end
    if TYPE == 3 || TYPE == 4
            subjfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic
shoulder movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE]; %store files
    end

        %folder to go to for images
    if separate == 1
        imgfolder = [subjfolder,'\sync_frames'];
    end
    if separate == 2  %images not separate from manual points
        imgfolder = subjfolder;
```

```
    end

    copyfile('fxn9.m',imgfolder)
    copyfile('fxn27.m',imgfolder)
    copyfile('fxn81.m',imgfolder)
    copyfile('fxnplot.m',imgfolder)



    cd(subjfolder)
    pickedpts=xlsread(namefile);

    cd(imgfolder)

    writefile = ['autopts2_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];  %title to
store auto spinepts
%     printRange = 'A2';

    %%%set by us; can be changed%%%
    X1 = 227.5;  Y1 = 172.5;  X2 = 966.5;  Y2 = 760.5;  %standard values for
cropping
    conversion=double(145.5); %convert excel value from cm to pix by
cm*conversion
    box81 = (81/2) - 0.5;  %using 81x81 box as start so 40 above&below
    box27 = (27/2) - 0.5;  %using 27x27 box as start so 13 above&below
    box9 = (9/2) - 0.5;  %using 9x9 box as start so 4 above&below
    EOI_x = (X2-X1)+1; EOI_y = (Y2-Y1)+1;  %end of image

    totalImgCutoff = 29; %for static trials, 34 for dynamic
    pixIntCutoff = 90; %for static trials, 115 for dynamic
    point30Cutoff = 68; %for static trials, 105 for dynamic

    values = zeros(framestop,15,'double');  %matrix of zeros for storing with
framestop rows
    %%%format of 'values' or excel file is 15 columns

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];

    %%%%CREATE 9 CF matrices
    CF_11 = [0.5,0.6,0.7;0.6,0.8,0.9;0.7,0.9,1];
    CF_12 = [0.6,0.5,0.6;0.7,0.8,0.7;0.9,1,0.9];
    CF_13 = [0.7,0.6,0.5;0.9,0.8,0.6;1,0.9,0.7];
    CF_21 = [0.5,0.7,0.9;0.6,0.8,1;0.5,0.7,0.9];
    CF_22 = [0.8,0.9,0.8;0.9,1,0.9;0.8,0.9,0.8];
    CF_23 = [0.9,0.7,0.5;1,0.8,0.6;0.9,0.7,0.5];
    CF_31 = [0.7,0.9,1;0.6,0.8,0.9;0.5,0.6,0.7];
    CF_32 = [0.9,1,0.9;0.7,0.8,0.7;0.6,0.5,0.6];
    CF_33 = [1,0.9,0.7;0.9,0.8,0.6;0.7,0.6,0.5];
    CF_t = [1,1,1;1,1,1;1,1,1];
```

```
    %for dynamic these 3 lines below are needed
    % framend = (framestop - framestart) + 1;
    % imgFrame = framestart - 1;
    %for i = 1:framend
%     c = 1;
    for i = framestart:framestop
        cd(imgfolder)  %go to folder with images
        frame = i
        %frame = i + imgFrame; in dynamic
        manual_xCM = double(pickedpts(i,1));
        manual_yCM = double(pickedpts(i,2));
        manual_x = double(pickedpts(i,1)*conversion);  %convert to pix
        manual_y = double(pickedpts(i,2)*conversion);
        manual_xy = manual_xCM + manual_yCM;
        if manual_xy < 1
            manID = 0;  %this frame was manually identified as no
identifiable scap border
        elseif manual_xy > 1
            manID = 1; %manual thought this frame had an identifiable scap
border
        end

        img_orig = imread([imagetitle,num2str(frame),'.jpeg']); %read in
image
        img2gray = rgb2gray(img_orig);  %convert to grayscale
        img = img2gray(Y1:Y2,X1:X2);  %crop image using standard values
        ROI_orig = img;  %create matrix of image intensities
        if manual_y > 0 && manual_x > 0
            pixInt = double(ROI_orig(round(manual_y),round(manual_x)));
%manPicked image intensity
        elseif manual_y <=0 || manual_x <= 0
            pixInt = 0;
        end

        %for first 5 frames use/get manual input
        if i < 5
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
%              c = c + 1;
            continue
        end

        j = i - 1;
        prevAutoX = values(j,2);
```

```
        prevAutoY = values(j,3);
        prevAutoXcm = values(j,6);
        prevAutoYcm = values(j,7);
        prevAuto_xy = prevAutoXcm + prevAutoYcm;

        if prevAuto_xy < 1
            %previously no identifiable, so use/get manual for this frame
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
        end

        %find overall image average, typical image avg under 30 doesnt have
an identifiable peak
        imgAvg = double(mean(mean(ROI_orig)));

        if imgAvg < totalImgCutoff
            %check if last picked was manual and identifiable
            if values(j,13) == 0
%                 if values(j,14) == 1
%                     %dont ask manual, keep moving
%                 end
            else
            %save as 0,0 i.e. auto thinks noID
            manOaut = 1; %auto used
            autoID = 0; autoXpix = 0; autoYpix = 0;
            dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
            dist_cm = dist_pix/conversion;
            pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame, img,PLANE);
%plot it
            continue
            end
        end

        %create ROI points - make 81x81box so last picked at center
```

```
        ROI81_X1 = prevAutoX - box81;  ROI81_X2 = prevAutoX + box81;
        ROI81_Y1 = prevAutoY - box81;  ROI81_Y2 = prevAutoY + box81;
        %for plotting ROI
        ROI = 81;
        plotX1 = ROI81_X1; plotY1 = ROI81_Y1; plotX2 = ROI81_X2; plotY2 =
ROI81_Y2;

        %make sure ROI doesnt extend passed image bounds
        if ROI81_X1 <= 0 || ROI81_Y1 <= 0 || ROI81_X2 > EOI_x || ROI81_Y2 >
EOI_y
            %ask for manual to put auto program back on track
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
        end


        if ROI == 81
            %call fxn81, give it 4ROI vals, origimg matrix, proper CF matrix
%             cd(mainfolder) %where you will find fxnROIs & fxnplot
            %return CF coords and corresponding 4tempROI vals
            [tX1_27,tX2_27,tY1_27,tY2_27,CFt1,maxAvg1] =
fxn81(ROI81_X1,ROI81_X2,ROI81_Y1,ROI81_Y2,ROI_orig,CF_22);
            %choose next CF matrix for ROI27 based on output
            if CFt1 == 11;
                CF1 = CF_11; ROI = 27;
            elseif CFt1 == 12;
                CF1 = CF_12; ROI = 27;
            elseif CFt1 == 13;
                CF1 = CF_13; ROI = 27;
            elseif CFt1 == 21;
                CF1 = CF_21; ROI = 27;
            elseif CFt1 == 22;
                CF1 = CF_22; ROI = 27;
            elseif CFt1 == 23;
                CF1 = CF_23; ROI = 27;
            elseif CFt1 == 31;
                CF1 = CF_31; ROI = 27;
            elseif CFt1 == 32;
                CF1 = CF_32; ROI = 27;
            elseif CFt1 == 33;
                CF1 = CF_33; ROI = 27;
            elseif CFt1 == 0  %%if coords returned from fxn81 are 0 (i.e
completely dark image)
```

```
                %save values as 0,0 i.e. auto thinks no id
                manOaut = 1; %auto used
                autoID = 0; autoXpix = 0; autoYpix = 0;
                dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
                dist_cm = dist_pix/conversion;
                pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
                values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%                cd(imgfolder)
                save('values.mat','values')   %save the matrix as matlab file
%                 cd(mainfolder) %where you will find fxnROIs & fxnplot
                fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame,
img,PLANE); %plot it
                continue
            end
        end  %end of if ROI = 81



        if ROI == 27
            [tX1_9,tX2_9,tY1_9,tY2_9,CFt2,maxAvg2] =
fxn27(tX1_27,tX2_27,tY1_27,tY2_27,ROI_orig,CF1);
            %choose next CF matrix for ROI9 based on output
            if CFt2 == 11;
                CF2 = CF_11;  ROI = 9;
            elseif CFt2 == 12;
                CF2 = CF_12;  ROI = 9;
            elseif CFt2 == 13;
                CF2 = CF_13;   ROI = 9;
            elseif CFt2 == 21;
                CF2 = CF_21;  ROI = 9;
            elseif CFt2 == 22;
                CF2 = CF_22; ROI = 9;
            elseif CFt2 == 23;
                CF2 = CF_23; ROI = 9;
            elseif CFt2 == 31;
                CF2 = CF_31; ROI = 9;
            elseif CFt2 == 32;
                CF2 = CF_32;  ROI = 9;
            elseif CFt2 == 33;
                CF2 = CF_33; ROI = 9;
            elseif CFt2 == 0  %%if coords returned from fxn27 are 0 (i.e
completely dark image)
                %save values as 0,0 i.e. auto thinks no id
                manOaut = 1; %auto used
                autoID = 0; autoXpix = 0; autoYpix = 0;
                dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
                dist_cm = dist_pix/conversion;
                pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
```

```
                values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%                cd(imgfolder)
                save('values.mat','values')   %save the matrix as matlab file
%                 cd(mainfolder) %where you will find fxnROIs & fxnplot
                fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame,
img,PLANE); %plot it
                continue
            end  %end of CFs
        end  %end of if ROI=27


        if ROI == 9
            [fX1,fY1,maxAvg3] = fxn9(tX1_9,tX2_9,tY1_9,tY2_9,ROI_orig,CF2);

            %check point 30 below to make sure it's dark
            point30belowY = fY1 + 30;
            point30belowX = fX1;
            point30below =
double(ROI_orig(round(point30belowY),round(point30belowX)));
            if point30below > point30Cutoff %if it's not dark below
% %                 make ROI27 around point30below
%                 tX1_27 = point30belowX - box27;  tX2_27 = point30belowX +
box27;
%                 tY1_27 = point30belowY - box27;  tY2_27 = point30belowY +
box27;
% %                 cal ROI27
%                 [tX1_9,tX2_9,tY1_9,tY2_9,CFt2,maxAvg2] =
fxn27(tX1_27,tX2_27,tY1_27,tY2_27,ROI_orig,CF_22);
% %                 call ROI9
%                 [fX1,fY1,maxAvg3] =
fxn9(tX1_9,tX2_9,tY1_9,tY2_9,ROI_orig,CF_t);
                if values(j,13) == 0
%                     if values(j,14) == 1
%                     %do nothing, keep moving
%                     end
                else
                manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
                values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
                save('values.mat','values')   %save the matrix as matlab file
                fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
                continue
                end
            end
```

```
            pixInt = double(ROI_orig(round(fY1),round(fX1)));  %pixel
intensity of point picked
        end %end of if ROI=9

        if pixInt < pixIntCutoff
            %check if last used manual
            if values(j,13) == 0
%                 if values(j,14) == 1
%                 %do nothing, keep moving
%                 end
            else
            %intensity of point picked too low, check manual
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
%             cd(imgfolder)
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnROIs & fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
            end
        end

        fX1cm = fX1/conversion;  fY1cm = fY1/conversion;

        %distance btwn coords of predicted and manual coords
        dist_pix = double(sqrt((fX1-manual_x).^2 + (fY1-manual_y).^2));
        dist_cm = dist_pix/conversion;
        manOaut = 1; %used manual
        autoID = 1;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
        values(i,:) =
[frame,double(fX1),double(fY1),manual_x,manual_y,double(fX1cm),double(fY1cm),
manual_xCM, manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%         cd(imgfolder)
        save('values.mat','values')   %save the matrix as matlab file
        %call function to plot manual and auto
%         cd(mainfolder) %where you will find fxnROIs & fxnplot
        fxnplot(plotX1, plotY1, plotX2, plotY2, manual_x, manual_y, fX1, fY1,
i,img,PLANE);




%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
```

```
        V = array2table(values,'VariableNames',{'Frame' 'autoX_pix'
'autoY_pix' 'manX_pix' 'manY_pix' 'autoX_cm' 'autoY_cm' 'manX_cm' 'manY_cm'
'Dist_pix' 'Dist_cm' 'PixIntensity' 'manOaut' 'manID' 'autoID'});
        %save to excel every 25 frames
        if i == 50
%             cd(imgfolder)
%             xlswrite(writefile,values,printRange)
            writetable(V,writefile)
        elseif mod(i,50) == 0
%             cd(imgfolder)
%             xlswrite(writefile,values,printRange)
            writetable(V,writefile)
        elseif i == framestop
%             cd(imgfolder)
            writetable(V,writefile)
%             xlswrite(writefile,values,printRange)
        end

        %print out what subject & frame we are on
        disp(['Processed subject ', int2str(SUBJ), ', COUNT
',int2str(COUNT),', frame ',int2str(frame)])

    end  %end of auto loop for frames start - stop
    V = array2table(values,'VariableNames',{'Frame' 'autoX_pix' 'autoY_pix'
'manX_pix' 'manY_pix' 'autoX_cm' 'autoY_cm' 'manX_cm' 'manY_cm' 'Dist_pix'
'Dist_cm' 'PixIntensity' 'manOaut' 'manID' 'autoID'});
    writetable(V,writefile)
% elapsedTime = toc
end %end of loop for all subjs

end  % end for type loop

elapsedTime = toc

end  %end of function
```

B.1 MATLAB CODE FOR CORRECTION FACTOR AND REGION SCANNING

B.1.1 ROI-81

```
function [tX1,tX2,tY1,tY2,CFt,maxAvg] = fxn81(X1,X2,Y1,Y2,orig_img,CF)

%split 81x81 into 9 27x27
```

```
y1 = Y1;      x1 = X1;
y27 = Y1+26;  x27 = X1+26;
y28 = Y1+27;  x28 = X1+27;
y54 = Y1+53;  x54 = X1+53;
y55 = Y1+54;  x55 = X1+54;
y81 = Y1+80;  x81 = X1+80;

ROI1 = zeros(3,3);
ROI1(1,1) = mean(mean(orig_img(y1:y27,x1:x27)));
ROI1(2,1) = mean(mean(orig_img(y28:y54,x1:x27)));
ROI1(3,1) = mean(mean(orig_img(y55:y81,x1:x27)));
ROI1(1,2) = mean(mean(orig_img(y1:y27,x28:x54)));
ROI1(2,2) = mean(mean(orig_img(y28:y54,x28:x54)));
ROI1(3,2) = mean(mean(orig_img(y55:y81,x28:x54)));
ROI1(1,3) = mean(mean(orig_img(y1:y27,x55:x81)));
ROI1(2,3) = mean(mean(orig_img(y28:y54,x55:x81)));
ROI1(3,3) = mean(mean(orig_img(y55:y81,x55:x81)));

%multiply ROI by CF matrix
ROI_CF = ROI1.*CF;

maxAvg1 = max(max(ROI_CF));  %find highest avg intensity
CFcoords = []; %for coordinates that have max avg
[CFcoords(:,1), CFcoords(:,2)] = find(ROI_CF == maxAvg1);
[M1,N1] = size(CFcoords);
% coords1 = [];

if M1 > 1
    %there is at least 1 tie
    %%%FIND A BETTER SOLUTION
    %for now: try last one
    coords1 = [CFcoords(M1,1),CFcoords(M1,2)];
elseif M1 == 9
    %there's probably no identifiable peak
    %choose 0,0 from here bcuz image is all same color
    %i.e. image is all dark
    coords1 = [0 0];
else
    coords1 = [CFcoords(1,1),CFcoords(1,2)];
end


if coords1(1,:) == [1 1]
    tY1 = y1; tY2 = y27; tX1 = x1; tX2 = x27; CFt = 11; maxAvg = ROI1(1,1);
elseif coords1(1,:) == [2 1]
    tY1 = y28; tY2 = y54; tX1 = x1; tX2 = x27; CFt = 21; maxAvg = ROI1(2,1);
elseif coords1(1,:) == [3 1]
    tY1 = y55; tY2 = y81; tX1 = x1; tX2 = x27; CFt = 31; maxAvg = ROI1(3,1);
elseif coords1(1,:) == [1 2]
    tY1 = y1; tY2 = y27; tX1 = x28; tX2 = x54; CFt = 12; maxAvg = ROI1(1,2);
elseif coords1(1,:) == [2 2]
    tY1 = y28; tY2 = y54; tX1 = x28; tX2 = x54; CFt = 22; maxAvg = ROI1(2,2);
elseif coords1(1,:) == [3 2]
    tY1 = y55; tY2 = y81; tX1 = x28; tX2 = x54; CFt = 32; maxAvg = ROI1(3,2);
```

```
elseif coords1(1,:) == [1 3]
    tY1 = y1; tY2 = y27; tX1 = x55; tX2 = x81; CFt = 13; maxAvg = ROI1(1,3);
elseif coords1(1,:) == [2 3]
    tY1 = y28; tY2 = y54; tX1 = x55; tX2 = x81; CFt = 23; maxAvg = ROI1(2,3);
elseif coords1(1,:) == [3 3]
    tY1 = y55; tY2 = y81; tX1 = x55; tX2 = x81; CFt = 33; maxAvg = ROI1(3,3);
elseif coords1(1,:) == [0 0]
    tY1 = 0; tY2 = 0; tX1 = 0; tX2 = 0; CFt = 0; maxAvg = ROI1(3,3);
end


% CFr = coords1(1,1); CFc = coords1(1,2);


end
```

B.1.2 ROI-27

```
function [tX1,tX2,tY1,tY2,CFt,maxAvg2] = fxn27(X1,X2,Y1,Y2,ROI_orig,CF)

%split 27x27 into 9 9x9
y1 = Y1;        x1 = X1;
y9 = Y1+8;      x9 = X1+8;
y10 = Y1+9;     x10 = X1+9;
y18 = Y1+17;    x18 = X1+17;
y19 = Y1+18;    x19 = X1+18;
y27 = Y1+26;    x27 = X1+26;

ROI2 = zeros(3);
ROI2(1,1) = mean(mean(ROI_orig(y1:y9,x1:x9)));
ROI2(2,1) = mean(mean(ROI_orig(y10:y18,x1:x9)));
ROI2(3,1) = mean(mean(ROI_orig(y19:y27,x1:x9)));
ROI2(1,2) = mean(mean(ROI_orig(y1:y9,x10:x18)));
ROI2(2,2) = mean(mean(ROI_orig(y10:y18,x10:x18)));
ROI2(3,2) = mean(mean(ROI_orig(y19:y27,x10:x18)));
ROI2(1,3) = mean(mean(ROI_orig(y1:y9,x19:x27)));
ROI2(2,3) = mean(mean(ROI_orig(y10:y18,x19:x27)));
ROI2(3,3) = mean(mean(ROI_orig(y19:y27,x19:x27)));

%multiply ROI by CF matrix
ROI_CF = ROI2.*CF;

maxAvg2 = max(max(ROI_CF));  %find highest avg intensitycoords3 = [];
CFcoords = [];
[CFcoords(:,1), CFcoords(:,2)] = find(ROI_CF == maxAvg2);
[M2,N2] = size(CFcoords);
```

```
if M2 > 1
    %there is at least 1 tie
    %%%FIND A BETTER SOLUTION
    %for now: try last one
    coords2 = [CFcoords(M2,1),CFcoords(M2,2)];
elseif M2 == 9
    %there's probably no identifiable peak
    %choose 0,0 from here bcuz image is all same color
    %i.e. image is all dark
    coords2 = [0 0];
else
    coords2 = [CFcoords(1,1),CFcoords(1,2)];
end

if coords2(1,:) == [1 1]
    tY1 = y1; tY2 = y9; tX1 = x1; tX2 = x9; CFt = 11; maxAvg2 = ROI2(1,1);
elseif coords2(1,:) == [2 1]
    tY1 = y10; tY2 = y18; tX1 = x1; tX2 = x9; CFt = 21; maxAvg2 = ROI2(2,1);
elseif coords2(1,:) == [3 1]
    tY1 = y19; tY2 = y27; tX1 = x1; tX2 = x9; CFt = 31; maxAvg2 = ROI2(3,1);
elseif coords2(1,:) == [1 2]
    tY1 = y1; tY2 = y9; tX1 = x10; tX2 = x18; CFt = 12; maxAvg2 = ROI2(1,2);
elseif coords2(1,:) == [2 2]
    tY1 = y10; tY2 = y18; tX1 = x10; tX2 = x18; CFt = 22; maxAvg2 =
ROI2(2,2);
elseif coords2(1,:) == [3 2]
    tY1 = y19; tY2 = y27; tX1 = x10; tX2 = x18; CFt = 32; maxAvg2 =
ROI2(3,2);
elseif coords2(1,:) == [1 3]
    tY1 = y1; tY2 = y9; tX1 = x19; tX2 = x27; CFt = 13; maxAvg2 = ROI2(1,3);
elseif coords2(1,:) == [2 3]
    tY1 = y10; tY2 = y18; tX1 = x19; tX2 = x27; CFt = 23; maxAvg2 =
ROI2(2,3);
elseif coords2(1,:) == [3 3]
    tY1 = y19; tY2 = y27; tX1 = x19; tX2 = x27; CFt = 33; maxAvg2 =
ROI2(3,3);
elseif coords2(1,:) == [0 0]
    tY1 = 0; tY2 = 0; tX1 = 0; tX2 = 0; CFt = 0; maxAvg2 = ROI2(3,3);
end



end %end of fxn27
```

B.1.3 ROI-9

```
function [fX1,fY1,maxAvg3] = fxn9(X1,X2,Y1,Y2,ROI_orig,CF)

%scan the 9x9 with moving avg
ROI3 = zeros(7,7);

c = 1;  %initialize gen counter
yc = 1;   xc = 1;  %position in ROI3 to store avg

for x = X1:1:(X1+6)
    for y = Y1:1:(Y1+6) %go til at row 7 so can scan 7-9 3x3
        xa = x+2; ya = y+2;  %where to end for scan
        ROI3(yc,xc) = mean(mean(ROI_orig(y:ya,x:xa)));
        c = c + 1;
        yc = yc + 1;
    end
    xc = xc + 1;
    yc = 1;
end

% %multiply ROI by CF matrix
% ROI_CF = ROI3.*CF;
%NO CF NEEDED IN THIS ONE

maxAvg = max(max(ROI3));  %find highest avg
coords = [];
[coords(:,1), coords(:,2)] = find(ROI3 == maxAvg);
[M3,N3] = size(coords);

if M3 > 1
    %there is at least 1 tie
    %%%FIND A BETTER SOLUTION
    %for now: try last one
    coords3 = [coords(M3,1),coords(M3,2)];
else
    coords3 = [coords(1,1),coords(1,2)];
end


if coords3(1,1) == 1
    tY1 = Y1;
elseif coords3(1,1) == 2
    tY1 = (Y1 + 1);
elseif coords3(1,1) == 3
    tY1 = (Y1 + 2);
elseif coords3(1,1) == 4
    tY1 = (Y1 + 3);
elseif coords3(1,1) == 5
    tY1 = (Y1 + 4);
elseif coords3(1,1) == 6
```

```
    tY1 = (Y1 + 5);
elseif coords3(1,1) == 7
    tY1 = (Y1 + 6);
else
end

if coords3(1,2) == 1
    tX1 = X1;
elseif coords3(1,2) == 2
    tX1 = (X1 + 1);
elseif coords3(1,2) == 3
    tX1 = (X1 + 2);
elseif coords3(1,2) == 4
    tX1 = (X1 + 3);
elseif coords3(1,2) == 5
    tX1 = (X1 + 4);
elseif coords3(1,2) == 6
    tX1 = (X1 + 5);
elseif coords3(1,2) == 7
    tX1 = (X1 + 6);
else
end

%final coords ***THESE VALS NEED TO BE CHECKED
%get final coords as center of the 3x3 moving avg
fX1 = tX1 + 1;
fY1 = tY1 + 1;
maxAvg3 = ROI_orig(round(fY1),round(fX1));
% maxAvg3 = maxAvg;




end    %end of fxn9
```

B.2 MATLAB CODE FOR CREATING DUPLICATE ULTRASAOUND IMAGE WITH PLOTS OF ROI-81, MANUAL AND AUTOMATED POINTS PICKED

```
function fxnplot(plotX1, plotY1, plotX2, plotY2, manual_x, manual_y, fX1,
fY1, i, img,plane)

clear g;
g=figure;
imshow(img);
hold on;
```

```
%plot ROI box (blue)
plot(plotX1, plotY1,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','b');
plot(plotX2, plotY1,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','b');
plot(plotX1, plotY2,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','b');
plot(plotX2, plotY2,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','b');
%plot manually picked pt (green)
plot(manual_x,manual_y,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','g');
%plot auto picked pt (magenta)
plot(fX1,fY1,'MarkerSize',20,'Marker','.','LineStyle','-
','MarkerEdgeColor','m');
%save
frame = i;
name=['picked2_',plane,'_',num2str(frame)];  %%%CHANGE FILENAME ENDING (i.e.
'front2')
saveas(g,name,'jpg')
close(g)
hold off;

end
```

APPENDIX C

MATLAB CODE FOR AUTOMATED PROGRAM FOR A DYNAMIC TRIAL

```
function CFchunk6_dyn

warning('off','all')
warning

clear
clc


%folder with autocode and ROI n fxn code n info
mainfolder = 'Z:\Protocols Boninger\Ultrasound and dynamic shoulder movement
(262)\Data (Analysis)\March Transfer\programs';

% TYPE = 1;  %1 is rest2, 2 is scap2, 5 is dynscap, 3 is front2, 4 is sag2
tic
for TYPE = 5
    disp(['TYPE ',num2str(TYPE)])
    cd(mainfolder)

if TYPE == 1
    PLANE = 'rest2';
    infofile = 'rest2_info.xlsx';
end
if TYPE == 2
    PLANE = 'scap2';
    infofile = 'scap2_info.xlsx';
end
if TYPE == 5
    PLANE = 'dynscap';
    infofile = 'dynscap_info.xlsx';
end
if TYPE == 3
    PLANE = 'front2';
    infofile = 'front2_info.xlsx';
```

```
end
if TYPE == 4
    PLANE = 'sag2';
    infofile = 'sag2_info.xlsx';
end


info = xlsread(infofile);
for a = 2:29        %num of subjects
%     tic
    SUBJ = info(a,1)
    valid = isnan(info(a,2));
    if valid == 1
        disp(['Did not process subject ', int2str(SUBJ)])
        continue
    end
    framestart = info(a,2);
    framestop = info(a,3);
    imageTitle = info(a,4);
    pointsTitle = info(a,5);
    separate = info(a,6);
    COUNT = info(a,7);

    if imageTitle == 1
        imagetitle = 'syncframe';
    elseif imageTitle == 2
        imagetitle = 'frame';
    end

    if pointsTitle == 1
        namefile = 'spine_points.xls';
    elseif pointsTitle == 2
        namefile = 'spine_points_ml.xls';
    elseif pointsTitle == 3
        namefile = 'spine_points_LS.xls';
    end

    if TYPE == 1 || TYPE == 2 || TYPE == 5
            subjfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic
shoulder movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE]; %store files
    end
    if TYPE == 3 || TYPE == 4
            subjfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic
shoulder movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE]; %store files
    end

        %folder to go to for images
    if separate == 1
        imgfolder = [subjfolder,'\sync_frames'];
    end
    if separate == 2  %images not separate from manual points
        imgfolder = subjfolder;
```

```
    end

    copyfile('fxn9.m',imgfolder)
    copyfile('fxn27.m',imgfolder)
    copyfile('fxn81.m',imgfolder)
    copyfile('fxnplot.m',imgfolder)


    cd(subjfolder)
    pickedpts=xlsread(namefile);

    cd(imgfolder)

    writefile = ['autopts2_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];  %title to
store auto spinepts
%     printRange = 'A2';

    %%%set by us; can be changed%%%
    X1 = 227.5;  Y1 = 172.5;  X2 = 966.5;  Y2 = 760.5;  %standard values for
cropping
    conversion=double(145.5); %convert excel value from cm to pix by
cm*conversion
    box81 = (81/2) - 0.5;  %using 81x81 box as start so 40 above&below
    box27 = (27/2) - 0.5;  %using 27x27 box as start so 13 above&below
    box9 = (9/2) - 0.5;  %using 9x9 box as start so 4 above&below
    EOI_x = (X2-X1)+1; EOI_y = (Y2-Y1)+1;  %end of image

    totalImgCutoff = 27; %for static trials, 34 for dynamic
    pixIntCutoff = 115; %for static trials, 115 for dynamic
    point30Cutoff = 76; %for static trials, 105 for dynamic

    values = zeros(framestop,15,'double');  %matrix of zeros for storing with
framestop rows
    %%%format of 'values' or excel file is 15 columns

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];

    %%%%CREATE 9 CF matrices
    CF_11 = [0.5,0.6,0.7;0.6,0.8,0.9;0.7,0.9,1];
    CF_12 = [0.6,0.5,0.6;0.7,0.8,0.7;0.9,1,0.9];
    CF_13 = [0.7,0.6,0.5;0.9,0.8,0.6;1,0.9,0.7];
    CF_21 = [0.5,0.7,0.9;0.6,0.8,1;0.5,0.7,0.9];
    CF_22 = [0.8,0.9,0.8;0.9,1,0.9;0.8,0.9,0.8];
    CF_23 = [0.9,0.7,0.5;1,0.8,0.6;0.9,0.7,0.5];
    CF_31 = [0.7,0.9,1;0.6,0.8,0.9;0.5,0.6,0.7];
    CF_32 = [0.9,1,0.9;0.7,0.8,0.7;0.6,0.5,0.6];
    CF_33 = [1,0.9,0.7;0.9,0.8,0.6;0.7,0.6,0.5];
    CF_t = [1,1,1;1,1,1;1,1,1];
```

```
%     for dynamic these 3 lines below are needed
    framend = (framestop - framestart) + 1;
    imgFrame = framestart - 1;

%     c = 1;
    for i = 1:framend
        cd(imgfolder)  %go to folder with images
%         frame = i
        frame = i + imgFrame   %in dynamic
        manual_xCM = double(pickedpts(i,1));
        manual_yCM = double(pickedpts(i,2));
        manual_x = double(pickedpts(i,1)*conversion);  %convert to pix
        manual_y = double(pickedpts(i,2)*conversion);
        manual_xy = manual_xCM + manual_yCM;
        if manual_xy < 1
            manID = 0;  %this frame was manually identified as no
identifiable scap border
        elseif manual_xy > 1
            manID = 1; %manual thought this frame had an identifiable scap
border
        end

        img_orig = imread([imagetitle,num2str(frame),'.jpeg']); %read in
image
        img2gray = rgb2gray(img_orig);  %convert to grayscale
        img = img2gray(Y1:Y2,X1:X2);  %crop image using standard values
        ROI_orig = img;  %create matrix of image intensities
        if manual_y > 0 && manual_x > 0
            pixInt = double(ROI_orig(round(manual_y),round(manual_x)));
%manPicked image intensity
        elseif manual_y <=0 || manual_x <= 0
            pixInt = 0;
        end

        %for first 5 frames use/get manual input
        if i < 5
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
%              c = c + 1;
            continue
        end

        j = i - 1;
        prevAutoX = values(j,2);
        prevAutoY = values(j,3);
```

```
        prevAutoXcm = values(j,6);
        prevAutoYcm = values(j,7);
        prevAuto_xy = prevAutoXcm + prevAutoYcm;

        if prevAuto_xy < 1
            %previously no identifiable, so use/get manual for this frame
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
        end

        %find overall image average, typical image avg under 30 doesnt have
an identifiable peak
        imgAvg = double(mean(mean(ROI_orig)));

        if imgAvg < totalImgCutoff
            %check if last picked was manual and identifiable
            if values(j,13) == 0
%                 if values(j,14) == 1
%                     %dont ask manual, keep moving
%                 end
            else
            %save as 0,0 i.e. auto thinks noID
            manOaut = 1; %auto used
            autoID = 0; autoXpix = 0; autoYpix = 0;
            dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
            dist_cm = dist_pix/conversion;
            pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame, img,PLANE);
%plot it
            continue
            end
        end

        %create ROI points - make 81x81box so last picked at center
        ROI81_X1 = prevAutoX - box81;  ROI81_X2 = prevAutoX + box81;
```

```
        ROI81_Y1 = prevAutoY - box81;  ROI81_Y2 = prevAutoY + box81;
        %for plotting ROI
        ROI = 81;
        plotX1 = ROI81_X1; plotY1 = ROI81_Y1; plotX2 = ROI81_X2; plotY2 =
ROI81_Y2;

        %make sure ROI doesnt extend passed image bounds
        if ROI81_X1 < 0 || ROI81_Y1 < 0 || ROI81_X2 > EOI_x || ROI81_Y2 >
EOI_y
            %ask for manual to put auto program back on track
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
        end


        if ROI == 81
            %call fxn81, give it 4ROI vals, origimg matrix, proper CF matrix
%             cd(mainfolder) %where you will find fxnROIs & fxnplot
            %return CF coords and corresponding 4tempROI vals
            [tX1_27,tX2_27,tY1_27,tY2_27,CFt1,maxAvg1] =
fxn81(ROI81_X1,ROI81_X2,ROI81_Y1,ROI81_Y2,ROI_orig,CF_22);
            %choose next CF matrix for ROI27 based on output
            if CFt1 == 11;
                CF1 = CF_11; ROI = 27;
            elseif CFt1 == 12;
                CF1 = CF_12; ROI = 27;
            elseif CFt1 == 13;
                CF1 = CF_13; ROI = 27;
            elseif CFt1 == 21;
                CF1 = CF_21; ROI = 27;
            elseif CFt1 == 22;
                CF1 = CF_22; ROI = 27;
            elseif CFt1 == 23;
                CF1 = CF_23; ROI = 27;
            elseif CFt1 == 31;
                CF1 = CF_31; ROI = 27;
            elseif CFt1 == 32;
                CF1 = CF_32; ROI = 27;
            elseif CFt1 == 33;
                CF1 = CF_33; ROI = 27;
            elseif CFt1 == 0  %%if coords returned from fxn81 are 0 (i.e
completely dark image)
                %save values as 0,0 i.e. auto thinks no id
```

```
                manOaut = 1; %auto used
                autoID = 0; autoXpix = 0; autoYpix = 0;
                dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
                dist_cm = dist_pix/conversion;
                pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
                values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%                cd(imgfolder)
                save('values.mat','values')   %save the matrix as matlab file
%                 cd(mainfolder) %where you will find fxnROIs & fxnplot
                fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame,
img,PLANE); %plot it
                continue
            end
        end  %end of if ROI = 81


        if ROI == 27
            [tX1_9,tX2_9,tY1_9,tY2_9,CFt2,maxAvg2] =
fxn27(tX1_27,tX2_27,tY1_27,tY2_27,ROI_orig,CF1);
            %choose next CF matrix for ROI9 based on output
            if CFt2 == 11;
                CF2 = CF_11;  ROI = 9;
            elseif CFt2 == 12;
                CF2 = CF_12;  ROI = 9;
            elseif CFt2 == 13;
                CF2 = CF_13;   ROI = 9;
            elseif CFt2 == 21;
                CF2 = CF_21;  ROI = 9;
            elseif CFt2 == 22;
                CF2 = CF_22; ROI = 9;
            elseif CFt2 == 23;
                CF2 = CF_23; ROI = 9;
            elseif CFt2 == 31;
                CF2 = CF_31; ROI = 9;
            elseif CFt2 == 32;
                CF2 = CF_32;  ROI = 9;
            elseif CFt2 == 33;
                CF2 = CF_33; ROI = 9;
            elseif CFt2 == 0  %%if coords returned from fxn27 are 0 (i.e
completely dark image)
                %save values as 0,0 i.e. auto thinks no id
                manOaut = 1; %auto used
                autoID = 0; autoXpix = 0; autoYpix = 0;
                dist_pix = double(sqrt((0-manual_x).^2 + (0-manual_y).^2));
                dist_cm = dist_pix/conversion;
                pixInt = 0;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
```

```
                values(i,:) =
[frame,autoXpix,autoYpix,manual_x,manual_y,0,0,manual_xCM,
manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%                cd(imgfolder)
                save('values.mat','values')   %save the matrix as matlab file
%                 cd(mainfolder) %where you will find fxnROIs & fxnplot
                fxnplot(0, 0, 0, 0, manual_x, manual_y, 0, 0, frame,
img,PLANE); %plot it
                continue
            end  %end of CFs
        end  %end of if ROI=27


        if ROI == 9
            [fX1,fY1,maxAvg3] = fxn9(tX1_9,tX2_9,tY1_9,tY2_9,ROI_orig,CF2);

            %check point 30 below to make sure it's dark
            point30belowY = fY1 + 30;
            point30belowX = fX1;
            point30below =
double(ROI_orig(round(point30belowY),round(point30belowX)));
            if point30below > point30Cutoff %if it's not dark below
% %                 make ROI27 around point30below
%                 tX1_27 = point30belowX - box27;  tX2_27 = point30belowX +
box27;
%                 tY1_27 = point30belowY - box27;  tY2_27 = point30belowY +
box27;
% %                 cal ROI27
%                 [tX1_9,tX2_9,tY1_9,tY2_9,CFt2,maxAvg2] =
fxn27(tX1_27,tX2_27,tY1_27,tY2_27,ROI_orig,CF_22);
% %                 call ROI9
%                 [fX1,fY1,maxAvg3] =
fxn9(tX1_9,tX2_9,tY1_9,tY2_9,ROI_orig,CF_t);
                if values(j,13) == 0
%                     if values(j,14) == 1
%                     %do nothing, keep moving
%                     end
                else
                manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
                values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
                save('values.mat','values')   %save the matrix as matlab file
                fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
                continue
                end
            end
```

```
            pixInt = double(ROI_orig(round(fY1),round(fX1)));  %pixel
intensity of point picked
        end %end of if ROI=9

        if pixInt < pixIntCutoff
            %check if last used manual
            if values(j,13) == 0
%                 if values(j,14) == 1
%                 %do nothing, keep moving
%                 end
            else
            %intensity of point picked too low, check manual
            manOaut = 0; %manual used

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
            values(i,:) =
[frame,double(manual_x),double(manual_y),manual_x,manual_y,double(manual_xCM)
,double(manual_yCM),manual_xCM,
manual_yCM,0,0,double(pixInt),manOaut,manID,manID];
%             cd(imgfolder)
            save('values.mat','values')   %save the matrix as matlab file
%             cd(mainfolder) %where you will find fxnROIs & fxnplot
            fxnplot(0, 0, 0, 0, manual_x, manual_y, manual_x, manual_y,
frame, img,PLANE); %plot it
            continue
            end
        end

        fX1cm = fX1/conversion;  fY1cm = fY1/conversion;

        %distance btwn coords of predicted and manual coords
        dist_pix = double(sqrt((fX1-manual_x).^2 + (fY1-manual_y).^2));
        dist_cm = dist_pix/conversion;
        manOaut = 1; %used manual
        autoID = 1;

%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
        values(i,:) =
[frame,double(fX1),double(fY1),manual_x,manual_y,double(fX1cm),double(fY1cm),
manual_xCM, manual_yCM,dist_pix,dist_cm,pixInt,manOaut,manID,autoID];
%         cd(imgfolder)
        save('values.mat','values')   %save the matrix as matlab file
        %call function to plot manual and auto
%         cd(mainfolder) %where you will find fxnROIs & fxnplot
        fxnplot(plotX1, plotY1, plotX2, plotY2, manual_x, manual_y, fX1, fY1,
i,img,PLANE);




%%%[frame#,autoXpix,autoYpix,manualXpix,manualYpix,autoXcm,autoYcm,manualXcm,
manualYcm,dist_pix,dist_cm,pix_int, manOaut, manID, autoID];
```

```
        V = array2table(values,'VariableNames',{'Frame' 'autoX_pix'
'autoY_pix' 'manX_pix' 'manY_pix' 'autoX_cm' 'autoY_cm' 'manX_cm' 'manY_cm'
'Dist_pix' 'Dist_cm' 'PixIntensity' 'manOaut' 'manID' 'autoID'});
        %save to excel every 25 frames
        if i == 50
%             cd(imgfolder)
%             xlswrite(writefile,values,printRange)
            writetable(V,writefile)
        elseif mod(i,50) == 0
%             cd(imgfolder)
%             xlswrite(writefile,values,printRange)
            writetable(V,writefile)
        elseif i == framestop
%             cd(imgfolder)
            writetable(V,writefile)
%             xlswrite(writefile,values,printRange)
        end

        %print out what subject & frame we are on
        disp(['Processed subject ', int2str(SUBJ), ',COUNT
',int2str(COUNT),', frame ',int2str(frame)])

    end  %end of auto loop for frames start - stop
    V = array2table(values,'VariableNames',{'Frame' 'autoX_pix' 'autoY_pix'
'manX_pix' 'manY_pix' 'autoX_cm' 'autoY_cm' 'manX_cm' 'manY_cm' 'Dist_pix'
'Dist_cm' 'PixIntensity' 'manOaut' 'manID' 'autoID'});
    writetable(V,writefile)
% elaspedTime = toc
end %end of loop for all subjs

end  % end for type loop

elapsedTime2 = toc

end  %end of function
```

APPENDIX D

MATLAB CODE FOR POST-PROCESSING

```
function evalCFc3

clear
clc

TYPE = 5;  %1 is rest2, 2 is scap2, 5 is dynscap, 3 is front2, 4 is sag2
if TYPE == 1
    PLANE = 'rest2';
    infofile = 'rest2_info.xlsx';
end
if TYPE == 2
    PLANE = 'scap2';
    infofile = 'scap2_info.xlsx';
end
if TYPE == 5
    PLANE = 'dynscap';
    infofile = 'dynscap_info.xlsx';
end
if TYPE == 3
    PLANE = 'front2';
    infofile = 'front2_info.xlsx';
end
if TYPE == 4
    PLANE = 'sag2';
    infofile = 'sag2_info.xlsx';
end


evals = zeros(29,16);
writefile = ['evalCFc3_',PLANE,'.xlsx'];
writefolder = ''Z:\Protocols Boninger\Ultrasound and dynamic shoulder
movement (262)\Data (Analysis)\March Transfer\post proc';
evalsAUTO = [];
writefileAUTO = ['evalAUTO_',PLANE,'.xlsx'];
evalsAUTOwmid = [];
```

```
writefileAUTOwmid = ['evalAUTOwmid_',PLANE,'.xlsx'];
evalsAUTOused = [];
writefileAUTOused = ['evalAUTOused_',PLANE,'.xlsx'];
evalsALL = [];
writefileALL = ['evalALL_',PLANE,'.xlsx'];
 c = 0;
 e = 0;  f = 0;
 d = 0;

info = xlsread(infofile);
for a = 1:29
    SUBJ = info(a,1)
    valid = isnan(info(a,2));
    if valid == 1
        disp(['Did not process subject ', int2str(SUBJ)])
        continue
    end

    separate = info(a,6);

        %folder to go to for autospinepts
    if separate == 1
        imgfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic shoulder
movement (262)\Data (Investigator)\US
Images\SCAP',num2str(SUBJ),'\',PLANE,'\sync_frames'];
    end
    if separate == 2  %images not separate from manual points
        imgfolder = ['Z:\Protocols Boninger\Ultrasound and dynamic shoulder
movement (262)\Data (Investigator)\US Images\SCAP',num2str(SUBJ),'\',PLANE];
    end

    cd(imgfolder)
    readfile = ['autopts2_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];  %file with
auto spinepts
    autopts = xlsread(readfile);
    lengthAuto = length(autopts);

    numImg = length(autopts);  %number of images for this subj

    manUsed = 0;
    manUsedID = 0;
    manNautoY = 0;
    manNID = 0;
    IDdist = [];
%     dist35 = 0;
    IDdistT = [];

    validYN = 0;

    tempD = [];
    usefultempD = [];
    usefultempD2 = [];
```

```
    for i = 1:numImg
        %how many times was manual input used
        if autopts(i,13) == 0
            manUsed = manUsed + 1;
        end

        %how many times manual input was used when image manually considered
        %identifiable (how many times did it actually need help)

        j = i - 1;

        if j > 0
            if (autopts(i,13) == 0) && (autopts(i,14) == 1) && (autopts(j,14)
~= 0)
                manUsedID = manUsedID + 1;
            end
        end

        %how many times manual said noID
        if autopts(i,14) == 0
            manNID = manNID + 1;
        end

        %how many times manual said noID but auto said identifiable
        if (autopts(i,14) == 0) && (autopts(i,15) == 1)
            manNautoY = manNautoY + 1;
        end

        %all the distances when both man & auto say identifiable
        if (autopts(i,14) == 1) && (autopts(i,15) == 1)
            IDdist(i,1) = autopts(i,10);
        end

        IDdistT(i,1) = autopts(i,10);

%         %how many times distance > 35 pix
%         if autopts(i,10) > 35
%             dist35 = dist35 + 1;
%         end

        k = i + 1;
        found = 0;



        if (autopts(i,15) == 1) &&(autopts(i,14) == 0)
            tempX1 = autopts(i,2);
            tempY1 = autopts(i,3);

            while j > 1  %check before
                if autopts(j,14) == 1
```

```
                tempX2 = autopts(j,4);
                tempY2 = autopts(j,5);
                tempD2 = ((tempX2 - tempX1).^2 + (tempY2-tempY1).^2);

                if tempD2 < 30
                    validYN = validYN + 1;
                    d = d + 1;
                    tempD(d,1) = tempD2;
                    d = d + 1;
                    found = 1;
                    break
                elseif tempD2 > 30
                    break
                end
            else
                j = j - 1;
            end
        end  %end while j>1

        if found == 0  %check after
            while k < lengthAuto
                if autopts(k,14) == 1
                    tempX2 = autopts(j,4);
                    tempY2 = autopts(j,5);
                    tempD2 = ((tempX2 - tempX1).^2 + (tempY2-tempY1).^2);

                    if tempD2 < 30
                        validYN = validYN + 1;
                        d = d + 1;
                        tempD(d,1) = tempD2;
                        d = d + 1;
                        break
                    elseif tempD2 > 30
                        break
                    end
                else
                    k = k + 1;
                end
            end
        elseif found == 1
            %dont check after just end
        end
    end   %end if auto says yes ID & man says noID


    if (autopts(i,13) == 1) && (autopts(i,14) == 1)  %if auto was used &
manually identifiable image
        c = c + 1;
        evalsAUTOwmid(c,1) = SUBJ;
        evalsAUTOwmid(c,2) = autopts(i,1);  %frame
        evalsAUTOwmid(c,3) = autopts(i,2);  %autoXpix
        evalsAUTOwmid(c,4) = autopts(i,3);
        evalsAUTOwmid(c,5) = autopts(i,4);  %manXpix
        evalsAUTOwmid(c,6) = autopts(i,5);
```

```
            evalsAUTOwmid(c,7) = autopts(i,6);  %autoXcm
            evalsAUTOwmid(c,8) = autopts(i,7);
            evalsAUTOwmid(c,9) = autopts(i,8);  %manXcm
            evalsAUTOwmid(c,10) = autopts(i,9);
            evalsAUTOwmid(c,11) = autopts(i,10); %distPIX
            evalsAUTOwmid(c,12) = autopts(i,11); %distCM
            evalsAUTOwmid(c,13) = sqrt((autopts(i,2)^2) + (autopts(i,3)^2));
%autoPTpix
            evalsAUTOwmid(c,14) = sqrt((autopts(i,4)^2) + (autopts(i,5)^2));
%manPTpix
        end


         if (autopts(i,13) == 1)  %if auto was used, regardless of
identifiable disagreements
            e = e + 1;
            evalsAUTOused(e,1) = SUBJ;
            evalsAUTOused(e,2) = autopts(i,1);  %frame
            evalsAUTOused(e,3) = autopts(i,2);  %autoXpix
            evalsAUTOused(e,4) = autopts(i,3);
            evalsAUTOused(e,5) = autopts(i,4);  %manXpix
            evalsAUTOused(e,6) = autopts(i,5);
            evalsAUTOused(e,7) = autopts(i,6);  %autoXcm
            evalsAUTOused(e,8) = autopts(i,7);
            evalsAUTOused(e,9) = autopts(i,8);  %manXcm
            evalsAUTOused(e,10) = autopts(i,9);
            evalsAUTOused(e,11) = autopts(i,10); %distPIX
            evalsAUTOused(e,12) = autopts(i,11); %distCM
            evalsAUTOused(e,13) = sqrt((autopts(i,2)^2) + (autopts(i,3)^2));
%autoPTpix
            evalsAUTOused(e,14) = sqrt((autopts(i,4)^2) + (autopts(i,5)^2));
%manPTpix
         end

            f = f + 1;
            evalsALL(f,1) = SUBJ;
            evalsALL(f,2) = autopts(i,1);  %frame
            evalsALL(f,3) = autopts(i,2);  %autoXpix
            evalsALL(f,4) = autopts(i,3);
            evalsALL(f,5) = autopts(i,4);  %manXpix
            evalsALL(f,6) = autopts(i,5);
            evalsALL(f,7) = autopts(i,6);  %autoXcm
            evalsALL(f,8) = autopts(i,7);
            evalsALL(f,9) = autopts(i,8);  %manXcm
            evalsALL(f,10) = autopts(i,9);
            evalsALL(f,11) = autopts(i,10); %distPIX
            evalsALL(f,12) = autopts(i,11); %distCM
            evalsALL(f,13) = sqrt((autopts(i,2)^2) + (autopts(i,3)^2));
%autoPTpix
            evalsALL(f,14) = sqrt((autopts(i,4)^2) + (autopts(i,5)^2));
%manPTpix

    end  %end of all images
```

```
    manUsedP = (manUsed/numImg)*100;
    manUsedIDP = (manUsedID/numImg)*100;
    manNIDP = (manNID/numImg)*100;
    manNautoYP = (manNautoY/manNID)*100;
    avgDistwID = mean(IDdist);
    avgDistT = mean(IDdistT);
%     tempL = length(tempD);
    avgDistVaYmN = mean(tempD);
    validAutoYManNT = validYN;
    validAutoYManNP = (validYN/manNautoY)*100;
    realManNAutoYP = ((manNautoY - validAutoYManNT)/manNID)*100;


    evals(a,:) = 
[SUBJ,numImg,manNID,manNIDP,manUsed,manUsedP,manUsedID,manUsedIDP,manNautoY,m
anNautoYP,validAutoYManNT,avgDistVaYmN,validAutoYManNP,realManNAutoYP,avgDist
wID,avgDistT];
    save('evals.mat','evals')
    E = array2table(evals,'VariableNames',{'SUBJ' 'TotalNumImgs' 
'TotalManNoID' 'Perc_ManNoID' 'AmtManIn' 'Perc_ManIn' 'AmtManInwID' 
'Perc_RealAmtManIn' 'manNautoY' 'Perc_manNautoY' 'VALIDmanNautoY' 
'avgDistVmanNautoY' 'Perc_VmanNautoY' 'Perc_ACTUALmanNAutoY' 'avgDistwID' 
'avgDistT'});

    save('evalsAUTO.mat','evalsAUTO')
    EA = array2table(evalsAUTO,'VariableNames',{'SUBJ' 'FRAME' 'autoXpix' 
'autoYpix' 'manXpix' 'manYpix' 'autoXcm' 'autoYcm' 'manXcm' 'manYcm' 
'distPIX' 'distCM'});

    save('evalsAUTOwmid.mat','evalsAUTOwmid')
    EAwmid = array2table(evalsAUTOwmid,'VariableNames',{'SUBJ' 'FRAME' 
'autoXpix' 'autoYpix' 'manXpix' 'manYpix' 'autoXcm' 'autoYcm' 'manXcm' 
'manYcm' 'distPIX' 'distCM' 'autoPTpix' 'manPTpix'});

    save('evalsAUTOused.mat','evalsAUTOused')
    EAused = array2table(evalsAUTOused,'VariableNames',{'SUBJ' 'FRAME' 
'autoXpix' 'autoYpix' 'manXpix' 'manYpix' 'autoXcm' 'autoYcm' 'manXcm' 
'manYcm' 'distPIX' 'distCM' 'autoPTpix' 'manPTpix'});

    save('evalsALL.mat','evalsALL')
    EAll = array2table(evalsALL,'VariableNames',{'SUBJ' 'FRAME' 'autoXpix' 
'autoYpix' 'manXpix' 'manYpix' 'autoXcm' 'autoYcm' 'manXcm' 'manYcm' 
'distPIX' 'distCM' 'autoPTpix' 'manPTpix'});

end  %end of for all subjs

cd(writefolder)
writetable(E,writefile)
writetable(EA,writefileAUTO)
writetable(EAwmid,writefileAUTOwmid)
writetable(EAused,writefileAUTOused)
writetable(EAll,writefileALL)
end  %end of function
```

APPENDIX E

RAW (EXPANDED) DATA TABLES FROM DATA PRE-PROCESSING

E.1 RAW DATA FOR PRE-PROCESSING OF MANUALLY IDENTIFIABLE IMAGES PER SUBJECT FOR EACH TESTING POSITION

E.1.1 Rest Testing Position

Testing Position: **Rest**					
		Variables			
Subject #	***Parameter***	**Total Image Pixel Intensity**	**Point Picked Pixel Intensity**	**Point 30 Below Pixel Intensity**	**Mean Distance between Coordinates of Identified Scapular Border**
1	*Mean ± stdev*	52.25 ± 5.3	121.88 ± 41.5	55.96 ± 14.6	11.24 ± 12.95
	Min, Max	38.93, 60.49	52, 240	23, 95	1, 88.14
2	*Mean ± stdev*	42.76 ± 2.7	93.54 ± 36.9	49.08 ± 14.2	15.42 ± 21.29
	Min, Max	32.28, 48.21	34, 248	16, 96	0, 195.23
3	*Mean ± stdev*	39.88 ± 2.6	130.95 ± 38.7	46.38 ± 18.2	12.60 ± 15.46
	Min, Max	34.37, 45.56	15, 222	19, 187	1, 94.37
4	*Mean ± stdev*	57.25 ± 3.6	131.92 ± 43.4	45.50 ± 11.7	12.94 ± 19.55

	Min, Max	49.17, 63.69	55, 249	20, 91	1, 190.77
5	*Mean ± stdev*	35.39 ± 2.2	155.39 ± 39.7	63.36 ± 14.6	11.57 ± 19.50
	Min, Max	31.12, 40.49	78, 250	26, 125	0, 202.56
6	*Mean ± stdev*	48.42 ± 4.5	144.86 ± 47.9	56.87 ± 15.3	13.12 ± 23.36
	Min, Max	35.41, 57.08	58, 252	25, 98	0, 234.55
7	*Mean ± stdev*	64.41 ± 6.2	169.36 ± 43.1	76.49 ± 11.7	13.01 ± 20.68
	Min, Max	44.03, 76.50	69, 254	46, 109	1, 192.04
8	*Mean ± stdev*	48.28 ± 4.7	146.46 ± 42.8	60.74 ± 12.5	9.99 ± 9.55
	Min, Max	37.77, 57.90	61, 249	31, 99	0, 75.33
9	*Mean ± stdev*	44.43 ± 4.4	142.13 ± 43.4	61.66 ± 16.5	9.87 ± 17.30
	Min, Max	30.34, 58.92	61, 254	17, 144	0, 230.36
10	*Mean ± stdev*	50.64 ± 7.3	114.79 ± 41.5	61.24 ± 17.8	12.89 ± 18.06
	Min, Max	28.06, 61.10	13, 238	19, 153	0, 197.20
11	*Mean ± stdev*	43.23 ± 8.3	176.10 ± 43.4	70.08 ± 15.1	5.63 ± 6.59
	Min, Max	28.26, 64.72	72, 254	32, 111	0, 58.08
12	*Mean ± stdev*	58.89 ± 4.1	151.40 ± 49.8	71.92 ± 17.0	10.81 ± 20.22
	Min, Max	47.30, 65.87	29, 254	22, 119	1, 271.49
13	*Mean ± stdev*	42.27 ± 11.4	172.59 ± 47.8	66.44 ± 20.9	20.31 ± 61.09
	Min, Max	27.47, 64.28	70, 255	19, 151	0, 427.57
14	*Mean ± stdev*	47.20 ± 5.5	185.81 ± 44.5	86.41 ± 20.8	9.50 ± 17.27
	Min, Max	35.69 58.80	61, 254	35, 164	0, 298.74
15	*Mean ± stdev*	55.25 ± 6.5	182.14 ± 45.3	89.29 ± 117.9	9.94 ± 18.17
	Min, Max	41.62, 70.17	79, 255	53, 173	0, 229.11
16	*Mean ± stdev*	62.20 ± 5.6	171.80 ± 44.1	76.19 ±15.7	10.31 ± 11.51
	Min, Max	46.94, 71.44	70, 254	37, 130	0, 137.52
AVERAGES	*Mean ± stdev*	49.55 ± 5.3	149.45 ± 43.4	64.85 ± 15.9	11.82 ± 19.53

	Min, Max	36.80, 60.33	54.81, 248.88	27.50, 127.81	0.31, 195.19

E.1.2 Static Scapular Plane Testing Position

Testing Position: **Static Humeral Elevation in the Scapular Plane**					
		Variables			
Subject #	***Parameter***	**Total Image Pixel Intensity**	**Point Picked Pixel Intensity**	**Point 30 Below Pixel Intensity**	**Mean Distance between Coordinates of Identified Scapular Border**
1	*Mean ± stdev*	46.91 ± 3.1	92.50 ± 38.4	51.80 ± 20.3	12.33 ± 19.3
	Min, Max	39.62, 53.08	12, 217	13, 103	0, 218.85
2	*Mean ± stdev*	38.71 ± 3.4	83.54 ± 41.1	38.80 ± 16.4	13.67 ± 28.3
	Min, Max	21.33, 45.11	13, 244	12, 99	0, 330.76
3	*Mean ± stdev*	35.11 ± 2.7	122.49 ± 58.1	46.66 ± 20.8	16.46 ± 23.1
	Min, Max	30.21, 40.14	13, 245	13, 80	0, 216.60
4	*Mean ± stdev*	53.90 ± 3.6	141.80 ± 65.0	45.96 ± 18.6	11.93 ± 21.1
	Min, Max	42.91, 63.27	14, 254	12, 145	1, 256.02
5	*Mean ± stdev*	40.72 ± 6.3	117.17 ± 49.1	43.53 ± 20.4	13.29 ± 20.3
	Min, Max	33.06, 53.33	33, 227	13, 82	0, 181.57
6	*Mean ± stdev*	43.55 ± 3.2	136.25 ± 45.3	50.09 ± 16.4	20.27 ± 26.3
	Min, Max	36.44, 49.81	16, 244	14, 81	1, 169.84
7	*Mean ± stdev*	57.97 ± 7.0	138.48 ± 34.3	60.42 ± 12.7	17.50 ± 32.0
	Min, Max	48.37, 72.22	58, 247	32, 97	0, 366.31
8	*Mean ± stdev*	44.17 ± 3.3	122.59 ± 58.7	50.66 ± 21.5	15.63 ± 22.0
	Min, Max	37.33, 53.00	13, 254	10, 92	0, 224.36
9	*Mean ± stdev*	45.40 ± 6.2	105.35 ± 39.0	51.75± 14.8	22.62 ± 49.0
	Min, Max	27.23, 59.73	34, 227	17, 92	0, 393.99
10	*Mean ± stdev*	44.68 ± 4.8	110.48 ± 48.4	56.46 ± 15.4	16.49 ± 18.7
	Min, Max	34.31, 51.35	26, 251	14, 111	0, 178.55
11	*Mean ± stdev*	38.39 ± 7.5	135.05 ± 58.8	56.40 ± 22.8	11.05 ± 17.7
	Min, Max	25.50, 52.14	24, 254	13, 125	0, 240.12
12	*Mean ± stdev*	60.40 ± 2.9	135.84 ± 62.3	61.95 ± 25.6	8.75 ± 7.9
	Min, Max	51.74, 68.26	42, 255	16, 159	0, 56.86
13	*Mean ± stdev*	43.62 ± 7.6	158.89 ± 45.1	52.07 ± 15.3	9.80 ± 15.8

	Min, Max	33.23, 62.77	64, 254	14, 114	0, 185.95
14	*Mean ± stdev*	54.60 ± 8.0	164.48 ± 52.4	69.23 ± 26.4	11.77 ± 19.2
	Min, Max	42.94, 71.03	38, 254	27, 164	1, 313.85
15	*Mean ± stdev*	58.10 ± 5.1	157.73 ± 66.8	67.44 ± 28.2	12.98 ± 17.6
	Min, Max	46.38, 65.95	40, 255	21, 151	0, 135.18
16	*Mean ± stdev*	58.41 ± 6.3	132.85 ± 60.7	55.01 ± 25.3	20.42 ± 31.8
	Min, Max	45.34, 71.09	14, 254	15, 137	0, 391.01
AVERAGES	*Mean ± stdev*	47.79 ± 5.1	128.47 ± 51.5	53.64 ± 20.1	14.69 ± 23.1
	Min, Max	37.25, 58.27	28.38, 246	16, 114.5	0.19, 241.24

E.1.3 Dynamic Scapular Plane Testing Position

Testing Position: **Dynamic Humeral Elevation in the Scapular Plane**					
		Variables			
Subject #	***Parameter***	**Total Image Pixel Intensity**	**Point Picked Pixel Intensity**	**Point 30 Below Pixel Intensity**	**Mean Distance between Coordinates of Identified Scapular Border**
1	*Mean ± stdev*	48.73 ± 6.4	118.14 ± 46.7	48.99 ± 15.1	17.71 ± 26.3
	Min, Max	31.71, 64.70	13, 254	13, 101	0, 340.26
2	*Mean ± stdev*	39.50 ± 4.2	102.37 ± 40.9	41.49 ± 15.5	16.36 ± 23.9
	Min, Max	27.21, 49.00	16, 252	8, 157	0, 322.63
3	*Mean ± stdev*	36.50 ± 5.2	125.23 ± 43.1	46.42 ± 17.7	14.91 ± 18.2
	Min, Max	23.41, 55.02	25, 254	9, 111	0, 317.06
4	*Mean ± stdev*	47.37 ± 5.8	124.95 ± 46.7	50.02 ± 14.6	17.16 ± 22.6
	Min, Max	25.20, 59.58	24, 254	9, 100	0,436.72
5	*Mean ± stdev*	59.56 ± 7.6	170.24 ± 40.9	72.16 ± 13.6	12.83 ± 14.4
	Min, Max	36.34, 77.35	54, 254	13, 124	0, 184.84
6	*Mean ± stdev*	44.59 ± 5.5	126.03 ± 49.2	52.09 ± 17.7	12.89 ± 14.5
	Min, Max	28.04, 62.17	30, 254	10, 111	0, 351.50
7	*Mean ± stdev*	62.64 ± 6.2	180.17 ± 48.6	75.62 ± 21.8	11.45 ± 13.2
	Min, Max	43.85, 75.93	45, 255	16, 187	0, 292.54
8	*Mean ± stdev*	40.92 ± 8.0	171.35 ± 38.9	55.97 ± 13.5	11.70 ± 11.8
	Min, Max	23.28, 64.81	71, 254	15, 106	0, 267.95

9	*Mean ± stdev*	51.96 ± 9.6	194.03 ± 42.8	86.81 ± 20.2	13.33 ± 14.0
	Min, Max	30.45, 72.76	45, 255	17, 217	0, 285.30
10	*Mean ± stdev*	61.32 ± 9.1	169.03 ± 47.4	78.13 ± 18.3	14.77 ± 15.0
	Min, Max	38.57, 86.10	33, 254	23, 158	0, 251.03
11	*Mean ± stdev*	57.25 ± 8.8	217.92 ± 31.4	103.78 ± 21.9	15.58 ± 16.1
	Min, Max	32.80, 83.16	96, 255	35, 192	0, 214.55
12	*Mean ± stdev*	49.15 ± 9.7	211.98 ± 39.5	88.16 ± 26.0	13.47 ± 18.3
	Min, Max	28.85, 81.81	40, 255	13, 179	0, 335.45
13	*Mean ± stdev*	50.78 ± 5.	120.70 ± 58.2	48.86 ± 18.2	13.00 ± 16.9
	Min, Max	34.67, 65.12	23, 254	11, 136	0, 281.80
14	*Mean ± stdev*	57.83 ± 9.8	203.46 ± 39.7	89.05 ± 19.6	14.06 ± 17.1
	Min, Max	29.50, 81.11	56, 255	28, 174	0, 436.56
AVERAGES	*Mean ± stdev*	50.58 ± 7.2	159.70 ± 43.8	66.97 ± 18.1	14.23 ± 17.3
	Min, Max	30.99, 69.90	40.79, 254.21	15.71, 146.64	0, 308.44

E.2 RAW DATA FOR PRE-PROCESSING OF IMAGES WITH NO IDENTIFIABLE SCAPULAR BORDER PER SUBJECT FOR EACH TESTING POSITION

Variable: **Total Image Intensity**				
Subject #	***Parameter***	Testing Position		
		Rest	**Static Humeral Elevation in Scapular Plane**	**Dynamic Humeral Elevation in Scapular Plane**
1	*Mean ± stdev*	47.94 ± 5.2	38.74 ± 6.9	46.35 ± 10.1
	Max	57.61	47.44	64.97
2	*Mean ± stdev*	31.75 ± 8.5	31.36 ± 8.2	36.54 ± 7.6
	Max	42.97	48.56	51.24
3	*Mean ± stdev*	35.29 ± 6.6	34.48 ± 5.2	35.12 ± 9.5
	Max	45.52	46.10	58.24

4	*Mean ± stdev*	48.08 ±12.1	47.12 ± 13.0	43.32 ± 9.5
	Max	61.24	68.24	58.67
5	*Mean ± stdev*	34.04 ± 2.7	39.39 ± 6.2	59.53 ± 13.3
	Max	40.19	52.25	77.62
6	*Mean ± stdev*	43.99 ± 8.7	40.43 ± 6.6	42.47 ± 6.3
	Max	59.68	49.84	59.01
7	*Mean ± stdev*	57.24 ±15.0	55.88 ± 9.2	58.28 ± 13.0
	Max	72.64	71.35	87.84
8	*Mean ± stdev*	38.01 ±8.9	44.50 ± 14.5	45.14 ± 14.8
	Max	50.31	78.27	81.23
9	*Mean ± stdev*	33.39 ±7.7	47.72 ± 19.5	45.48 ± 19.0
	Max	41.98	77.44	92.37
10	*Mean ± stdev*	30.96 ±4.6	32.78 ± 8.0	57.81 ± 13.9
	Max	48.25	44.01	98.52
11	*Mean ± stdev*	22.90 ±3.5	34.17 ± 8.9	43.90 ± 19.0
	Max	28.77	44.36	83.32
12	*Mean ± stdev*	53.73 ±8.1	60.51 ± 14.4	39.20 ± 17.3
	Max	70.58	89.20	108.30
13	*Mean ± stdev*	39.98 ±13.4	40.03 ± 10.7	45.95 ± 11.8
	Max	64.01	70.24	60.86
14	*Mean ± stdev*	42.57 ±10.1	53.16 ± 14.9	49.80 ± 16.0
	Max	58.10	75.99	81.40
15	*Mean ± stdev*	52.55 ±9.6	52.46 ± 9.5	
	Max	70.39	65.48	
16	*Mean ± stdev*	50.03 ±16.4	52.67 ± 10.4	
	Max	70.55	66.63	
AVERGAES	*Mean ± stdev*	41.40 ± 8.8	44.09 ± 10.4	46.35 ± 12.9
	Max	55.17	62.21	75.97

APPENDIX F

RAW (EXTENDED) DATA TABLES FOR DATA POST-PROCESSING RESULTS FROM EACH TESTING POSITION

F.1 TESTING POSITION: REST

Subj.#	Total Num Imgs	Num Imgs Man. ID'd	Num Imgs Auto ID'd	Num Imgs Man = Y, Auto = N	Num Imgs Man. Not ID'd	Num Imgs Auto Not ID'd	Num Imgs Man = N, Auto = Y	Total Num Imgs Manual Input Requested	Num Imgs Manual Input Used for ID'd Img
1	371	179	216	0	192	155	37	191	37
2	469	405	412	0	64	57	7	167	110
3	318	148	177	0	170	141	29	186	46
4	386	225	263	0	161	123	38	152	29
5	316	181	197	0	135	119	16	195	76
6	253	148	171	0	105	82	23	125	43
7	416	271	291	0	145	125	20	253	128
8	240	198	206	0	42	34	8	110	77
9	315	288	288	0	27	27	0	148	122
10	300	226	226	1	74	74	1	158	85
11	337	319	317	2	18	20	0	151	133

12	436	248	261	0	188	175	13	282	107
13	532	392	403	6	140	129	17	267	145
14	535	472	479	0	63	56	7	268	213
15	388	202	214	0	186	174	12	276	102
16	531	432	444	0	99	87	12	256	169
17	676	480	508	0	196	168	28	386	218
18	554	336	358	0	218	196	22	372	176
19	555	469	472	0	86	83	3	307	224
20	712	644	654	1	68	58	11	367	310
21	715	660	665	0	55	50	5	375	326
22	639	442	448	0	197	191	6	414	224
23	673	548	552	0	125	121	4	378	257
24	585	539	540	0	46	45	1	301	256
25	756	669	696	0	87	60	27	257	197
26	642	588	598	0	54	44	10	332	289
27	622	524	529	0	98	93	5	356	263
28	742	480	505	0	262	237	25	446	209
29	712	689	689	0	23	23	0	362	339
Mean	507.79	393.17	406.17	0.34	114.62	101.62	13.34	270.28	169.31

Abbreviations:

- Subj.# =
- Total Num Imgs =
- Num Imgs Man. ID’d
- Num Imgs Auto ID’d
- Num Imgs Man = Y, Auto = N
- Num Imgs Man. Not ID’d
- Num Imgs Auto Not ID’d
- Num Imgs Man = N, Auto = Y
- Total Num Imgs Manual Input Requested
- Num Imgs Manual Input Used for ID’d Img

F.2 TESTING POSITION: STATIC HUMERAL ELEVATION IN THE SCAPULAR PLANE

Subj.#	Total Num Imgs	Num Imgs Man. ID'd	Num Imgs Auto ID'd	Num Imgs Man = Y, Auto = N	Num Imgs Man. Not ID'd	Num Imgs Auto Not ID'd	Num Imgs Man = N, Auto = Y	Total Num Imgs Manual Input Requested	Num Imgs Manual Input Used for ID'd Img
1	421	313	322	0	108	99	9	198	100
2	374	283	283	1	91	91	1	196	107
3	445	197	224	0	248	221	27	296	75
4	274	179	181	0	95	93	2	116	23
5	251	142	160	0	109	91	18	135	44
6	287	125	170	0	162	117	45	150	33
7	312	163	199	0	149	113	36	153	40
8	302	220	232	0	82	70	12	150	80
9	229	200	205	1	29	24	6	75	52
10	261	239	240	0	22	21	1	104	84
11	367	315	299	17	52	68	1	157	108
12	510	353	370	0	157	140	17	246	106
13	533	414	422	0	119	111	8	202	91
14	432	343	374	0	89	58	31	155	97
15	429	291	309	0	138	120	18	200	80
16	470	308	331	0	162	139	23	215	76
17	542	355	392	0	187	150	37	251	101
18	524	459	462	0	65	62	3	235	173
19	619	395	412	0	224	207	17	388	181
20	690	604	609	0	86	81	5	268	188

21	535	494	494	0	41	41	0	203	162
22	619	557	559	0	62	60	2	333	273
23	677	435	461	0	242	216	26	381	165
24	439	312	316	0	127	123	4	248	125
25	724	540	566	0	184	158	26	285	127
26	648	598	600	0	50	48	2	295	247
27	600	472	489	0	128	111	17	267	156
28	683	518	546	0	165	137	28	321	184
29	658	599	603	0	59	55	4	291	237
Mean	477.76	359.41	373.45	0.66	118.34	104.31	14.69	224.62	121.21

F.3 TESTING POSITION: STATIC HUMERAL ELEVATION IN THE FRONTAL PLANE

Subj.#	**Total Num Imgs**	**Num Imgs Man. ID'd**	**Num Imgs Auto ID'd**	**Num Imgs Man = Y, Auto = N**	**Num Imgs Man. Not ID'd**	**Num Imgs Auto Not ID'd**	**Num Imgs Man = N, Auto = Y**	**Total Num Imgs Manual Input Requested**	**Num Imgs Manual Input Used for ID'd Img**
1	360	252	254	0	108	106	2	161	55
2	430	309	316	0	121	114	7	207	94
3	549	220	246	10	329	303	36	376	83
4	381	168	185	0	213	196	17	230	34
5	259	159	168	0	100	91	9	124	33
6	309	170	191	2	139	118	23	170	55
7	269	193	207	0	76	62	14	124	62

8	223	168	180	0	55	43	12	100	58
9	263	201	204	3	62	59	6	126	70
10	213	179	182	0	34	31	3	82	52
11	428	391	382	11	37	46	2	173	139
12	452	299	328	0	153	124	29	203	79
13	446	293	303	5	153	143	15	232	94
14	362	253	270	0	109	92	17	188	97
15	353	253	260	0	100	93	7	167	74
16	577	442	476	0	135	101	34	210	109
17	660	438	467	0	222	193	29	347	154
18	442	401	407	0	41	35	6	193	159
19	655	509	541	0	146	114	32	328	214
20	624	505	516	0	119	108	11	341	234
21	607	552	554	0	55	53	2	317	264
22	648	546	555	0	102	93	9	369	276
23	662	340	365	0	322	297	25	413	116
24	762	592	615	0	170	147	23	335	188
25	605	412	442	0	193	163	30	255	92
26	687	536	567	0	151	120	31	318	199
27	782	689	697	0	93	85	8	363	278
28	684	406	442	0	278	242	36	414	172
29	744	691	693	0	53	51	2	368	318
Mean	497.79	364.38	379.76	1.07	133.41	118.03	16.44827586	249.45	132.83

F.4 TESTING POSITION: STATIC HUMERAL ELEVATION IN THE SAGITTAL PLANE

Subj.#	Total Num Imgs	Num Imgs Man. ID'd	Num Imgs Auto ID'd	Num Imgs Man = Y, Auto = N	Num Imgs Man. Not ID'd	Num Imgs Auto Not ID'd	Num Imgs Man = N, Auto = Y	Total Num Imgs Manual Input Requested	Num Imgs Manual Input Used for ID'd Img
1	403	330	338	0	73	65	8	162	97
2	429	365	371	0	64	58	6	218	162
3	415	138	173	0	277	242	35	288	46
4	322	143	166	0	179	156	23	183	28
5	286	209	221	0	77	65	12	123	58
6	277	141	193	0	136	84	52	127	43
7	304	152	177	0	152	127	25	190	63
8	299	209	220	0	90	79	11	153	74
9	255	161	168	1	94	87	8	143	58
10	208	190	191	0	18	17	1	75	59
11	301	260	256	4	41	45	0	138	97
12	563	366	385	0	197	178	19	292	114
13	421	265	280	0	156	141	15	195	54
14	246	191	197	0	55	49	6	137	89
15	367	278	289	0	89	78	11	169	92
16	486	321	347	0	165	139	26	223	84
17	561	431	450	0	130	111	19	249	138
18	515	469	473	0	46	42	4	249	207
19	608	481	497	0	127	111	16	317	206

20	536	443	449	0	93	87	6	293	207
21	639	602	604	0	37	35	2	256	221
22	612	514	518	0	98	94	4	347	254
23	547	242	318	0	305	229	76	320	92
24	602	481	498	0	121	104	17	248	144
25	682	442	461	0	240	221	19	318	98
26	617	511	553	0	106	64	42	237	173
27	579	471	490	0	108	89	19	265	177
28	705	459	480	0	246	225	21	422	197
29	664	637	638	0	27	26	1	316	290
Mean	463.76	341.45	358.66	0.17	122.31	105.10	17.38	229.41	124.90

F.5 TESTING POSITION: DYNAMIC HUMERAL ELEVATION IN THE SCAPULAR PLANE

Subj.#	**Total Num Imgs**	**Num Imgs Man. ID’d**	**Num Imgs Auto ID’d**	**Num Imgs Man = Y, Auto = N**	**Num Imgs Man. Not ID’d**	**Num Imgs Auto Not ID’d**	**Num Imgs Man = N, Auto = Y**	**Total Num Imgs Manual Input Requested**	**Num Imgs Manual Input Used for ID’d Img**
1	4419	2080	2402	0	2339	2017	322	2549	533
2	4332	2781	2955	0	1551	1377	174	2473	1099
3	4460	3555	3646	44	905	814	135	1987	1228
4	4451	2815	3091	1	1636	1360	277	1993	636
5	4445	3545	3736	0	900	709	191	1835	1126
6	4453	4035	4102	0	418	351	67	1666	1316
7	4457	3476	3686	0	981	771	210	2037	1268

8	4419	4183	4150	62	236	269	29	1199	996
9	4460	4233	4243	0	227	217	10	2013	1796
10	4423	3325	3428	0	1098	995	103	2295	1301
11	4560	3930	4012	0	630	548	82	2485	1937
12	4428	4057	4083	0	371	345	26	2043	1705
13	4438	3499	3699	0	939	739	200	1768	1031
14	4439	3967	4041	0	472	398	74	2132	1734
Mean	4441.71	3534.36	3662.43	7.64	907.36	779.29	135.71	2033.93	1264.71

APPENDIX G

MATLAB CODE FOR INTERACTIVE AUTOMATED POINT SELECTION PROGRAM TO BE USED DURING DATA PROCESSING (WITHOUT PRE-SELECTED MANUAL POINTS)

```
function autoPts
%this file needs to be stored and run in the same folder with
%code for fxn81,27,9,plot, etc
%mainfolder = '...';

%try adding keyboard function to stop function

tic

%disable warnings
warning('off','all')
warning

%clear contents
clear
clc


%GET FROM CMD LINE
%get subject number
subj1 = input('Please enter subject number (i.e. "4"). \n', 's');
SUBJ = str2num(subj1);
%get folder for subjects
disp('Please select the subject folder')
disp('(i.e. Z:\Protocols Boninger\...\SCAP4)')
subjfolder = uigetdir;
%get testing position/plane
PLANE = input('What is the testing position? (i.e. "rest2" or "dynscap") \n',
's');
```

```
%get framestart
framestart1 = input('Please enter the frame number to start with. \n', 's');
framestart = str2num(framestart1);
%get framestop
framestop1 = input('Please enter the last frame number. \n', 's');
framestop = str2num(framestop1);
%get image title (i.e. syncframe or frame)
imagetitle = input('Please enter the title of the images for this subject \n
(i.e. "syncframe" or "frame"). \n', 's');
%get folder with images
disp('Please select the folder where the images are stored')
disp('(i.e. Z:\Protocols Boninger\...\SCAP4\rest2 or
Z:\...\rest2\sync_frames) ')
imgfolder = uigetdir;


copyfile('fxn9.m',imgfolder)
copyfile('fxn27.m',imgfolder)
copyfile('fxn81.m',imgfolder)
copyfile('fxnplotA.m',imgfolder)


writefile = ['AUTOPTS_SCAP',num2str(SUBJ),'_',PLANE,'.xlsx'];  %title to
store auto pts

%%%set by us; can be changed%%%
X1 = 227.5;  Y1 = 172.5;  X2 = 966.5;  Y2 = 760.5;  %standard values for
cropping
conversion=double(145.5); %convert excel value from cm to pix by
cm*conversion
box81 = (81/2) - 0.5;  %using 81x81 box as start so 40 above&below
box27 = (27/2) - 0.5;  %using 27x27 box as start so 13 above&below
box9 = (9/2) - 0.5;  %using 9x9 box as start so 4 above&below
EOI_x = (X2-X1)+1; EOI_y = (Y2-Y1)+1;  %end of image

%cutoff values for static trials
totalImgCutoff = 29;
pixIntCutoff = 90;
point30Cutoff = 68;


%matrix of zeros for pre-allocating values matrix with number of images
blanks = framestop - framestart + 1;
values = zeros(blanks,9,'double');
%%%format of 'values' or excel file is 9 columns
%%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int, manOaut,reason];

%%%%CREATE 9 CF matrices
CF_11 = [0.5,0.6,0.7;0.6,0.8,0.9;0.7,0.9,1];
CF_12 = [0.6,0.5,0.6;0.7,0.8,0.7;0.9,1,0.9];
CF_13 = [0.7,0.6,0.5;0.9,0.8,0.6;1,0.9,0.7];
CF_21 = [0.5,0.7,0.9;0.6,0.8,1;0.5,0.7,0.9];
CF_22 = [0.8,0.9,0.8;0.9,1,0.9;0.8,0.9,0.8];
```

```
CF_23 = [0.9,0.7,0.5;1,0.8,0.6;0.9,0.7,0.5];
CF_31 = [0.7,0.9,1;0.6,0.8,0.9;0.5,0.6,0.7];
CF_32 = [0.9,1,0.9;0.7,0.8,0.7;0.6,0.5,0.6];
CF_33 = [1,0.9,0.7;0.9,0.8,0.6;0.7,0.6,0.5];
CF_t = [1,1,1;1,1,1;1,1,1];



sync = framestart + 5;


%start running auto for all images
for i = framestart:framestop
    cd(imgfolder)  %go to folder with images
    frame = i

    img_orig = imread([imagetitle,num2str(frame),'.jpeg']); %read in image
    img2gray = rgb2gray(img_orig);  %convert to grayscale
    img = img2gray(Y1:Y2,X1:X2);  %crop image using standard values
    ROI_orig = img;  %create matrix of image intensities

    %for first 5 frames use/get manual input
    if i < sync
        manOaut = 0; %manual used, 0 for manual, 1 for auto
        reason = 1 %1 = sync
        disp('HELP! Please select scapular border')
        imshow(img); hold on
        [fX1,fY1] = ginput(1);   %get manual input
        hold off
        close all
        fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
        plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
        fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
        close all
        if fY1 > 0 && fX1 > 0
            pixInt = double(ROI_orig(round(fY1),round(fX1)));
        else
            pixInt = 0;
        end
        %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
        values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
        continue
    end

    j = i - 1;
    prevAutoX = values(j,3);
    prevAutoY = values(j,4);
    prevAutoXcm = values(j,5);
    prevAutoYcm = values(j,6);
    prevAuto_xy = prevAutoXcm + prevAutoYcm;
```

```
    if prevAuto_xy < 1
        manOaut = 0; %manual used, 0 for manual, 1 for auto
%          reason = 0; %let last reason remain the same
        disp('HELP! Please select scapular border')
        imshow(img); hold on
        [fX1,fY1] = ginput(1);   %get manual input
        hold off
        close all
        fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
        plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
        fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
        close all
        if fY1 > 0 && fX1 > 0
            pixInt = double(ROI_orig(round(fY1),round(fX1)));
        else
            pixInt = 0;
        end
        %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
        values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
        continue
    end

    %find overall image average, typical image avg under 30 doesnt have an
identifiable peak
    imgAvg = double(mean(mean(ROI_orig)));

    if imgAvg < totalImgCutoff
        manOaut = 1; %auto used
        reason = 2 %2 is too dark
        fX1 = 0; fY1 = 0;
        fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
        pixInt = 0;
        plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
        %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
        values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
        save('values.mat','values')   %save the matrix as matlab file
        fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
        continue
    end

    %create ROI points - make 81x81box so last picked at center
    ROI81_X1 = prevAutoX - box81;  ROI81_X2 = prevAutoX + box81;
    ROI81_Y1 = prevAutoY - box81;  ROI81_Y2 = prevAutoY + box81;
    %for plotting ROI
    ROI = 81;
    plotX1 = ROI81_X1; plotY1 = ROI81_Y1; plotX2 = ROI81_X2; plotY2 =
ROI81_Y2;
```

```
    %make sure ROI doesnt extend passed image bounds
    if ROI81_X1 <= 0 || ROI81_Y1 <= 0 || ROI81_X2 > EOI_x || ROI81_Y2 > EOI_y
        %ask for manual to put auto program back on track
        manOaut = 0; %manual used, 0 for manual, 1 for auto
        reason = 3 %3 = ROI81 out of bounds
        disp('HELP! Please select scapular border')
        imshow(img); hold on
        [fX1,fY1] = ginput(1);   %get manual input
        hold off
        close all
        fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
        plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
        fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
        close all
        if fY1 > 0 && fX1 > 0
            pixInt = double(ROI_orig(round(fY1),round(fX1)));
        else
            pixInt = 0;
        end
        %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
        values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
        continue
    end


    if ROI == 81
        %call fxn81, give it 4ROI vals, origimg matrix, proper CF matrix
        %return CF coords and corresponding 4tempROI vals
        [tX1_27,tX2_27,tY1_27,tY2_27,CFt1,maxAvg1] =
fxn81(ROI81_X1,ROI81_X2,ROI81_Y1,ROI81_Y2,ROI_orig,CF_22);
        %choose next CF matrix for ROI27 based on output
        if CFt1 == 11;
            CF1 = CF_11; ROI = 27;
        elseif CFt1 == 12;
            CF1 = CF_12; ROI = 27;
        elseif CFt1 == 13;
            CF1 = CF_13; ROI = 27;
        elseif CFt1 == 21;
            CF1 = CF_21; ROI = 27;
        elseif CFt1 == 22;
            CF1 = CF_22; ROI = 27;
        elseif CFt1 == 23;
            CF1 = CF_23; ROI = 27;
        elseif CFt1 == 31;
            CF1 = CF_31; ROI = 27;
        elseif CFt1 == 32;
            CF1 = CF_32; ROI = 27;
        elseif CFt1 == 33;
            CF1 = CF_33; ROI = 27;
```

```
        elseif CFt1 == 0  %%if coords returned from fxn81 are 0 (i.e
completely dark image)
            %save values as 0,0 i.e. auto thinks no id
            manOaut = 1; %auto used
            reason = 4 %4 too dark in ROI
            fX1 = 0; fY1 = 0;
            fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
            pixInt = 0;
            plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
            %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
            values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
            save('values.mat','values')   %save the matrix as matlab file
            fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
            continue
        end
    end  %end of if ROI = 81


    if ROI == 27
        [tX1_9,tX2_9,tY1_9,tY2_9,CFt2,maxAvg2] =
fxn27(tX1_27,tX2_27,tY1_27,tY2_27,ROI_orig,CF1);
        %choose next CF matrix for ROI9 based on output
        if CFt2 == 11;
            CF2 = CF_11;  ROI = 9;
        elseif CFt2 == 12;
            CF2 = CF_12;  ROI = 9;
        elseif CFt2 == 13;
            CF2 = CF_13;   ROI = 9;
        elseif CFt2 == 21;
            CF2 = CF_21;  ROI = 9;
        elseif CFt2 == 22;
            CF2 = CF_22; ROI = 9;
        elseif CFt2 == 23;
            CF2 = CF_23; ROI = 9;
        elseif CFt2 == 31;
            CF2 = CF_31; ROI = 9;
        elseif CFt2 == 32;
            CF2 = CF_32;  ROI = 9;
        elseif CFt2 == 33;
            CF2 = CF_33; ROI = 9;
        elseif CFt2 == 0  %%if coords returned from fxn27 are 0 (i.e
completely dark image)
            %save values as 0,0 i.e. auto thinks no id
            manOaut = 1; %auto used
            reason = 2 %2 is too dark
            fX1 = 0; fY1 = 0;
            fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
            pixInt = 0;
            plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
            %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
```

```
            values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
            save('values.mat','values')   %save the matrix as matlab file
            fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
            continue
        end  %end of CFs
    end  %end of if ROI=27


    if ROI == 9
        [fX1,fY1,maxAvg3] = fxn9(tX1_9,tX2_9,tY1_9,tY2_9,ROI_orig,CF2);

        %check point 30 below to make sure it's dark
        point30belowY = fY1 + 30;
        point30belowX = fX1;
        point30below =
double(ROI_orig(round(point30belowY),round(point30belowX)))

        if point30below > point30Cutoff %if it's not dark below, get manual
            manOaut = 0; %manual used, 0 for manual, 1 for auto
            reason = 5 %5 = point 30 below is too dark
            disp('HELP! Please select scapular border')
            imshow(img); hold on
            [fX1,fY1] = ginput(1);   %get manual input
            hold off
            close all
            fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
            plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
            fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
            close all
            if fY1 > 0 && fX1 > 0
                pixInt = double(ROI_orig(round(fY1),round(fX1)));
            else
                pixInt = 0;
            end
            %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
            values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
            continue
        end

        pixInt = double(ROI_orig(round(fY1),round(fX1)));  %pixel intensity
of point picked
    end %end of if ROI=9

    if pixInt < pixIntCutoff
        %intensity of point picked too low, get manual
        manOaut = 0; %manual used, 0 for manual, 1 for auto
        reason = 6 %6 = point picked not bright enough
        disp('HELP! Please select scapular border')
        imshow(img); hold on
```

```
        [fX1,fY1] = ginput(1);   %get manual input
        hold off
        close all
        fX1_cm = fX1/conversion; fY1_cm = fY1/conversion;
        plotX1 = 0; plotY1 = 0; plotX2 = 0; plotY2 = 0;  %ROI81
        fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)
        close all
        if fY1 > 0 && fX1 > 0
            pixInt = double(ROI_orig(round(fY1),round(fX1)));
        else
            pixInt = 0;
        end
        %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
        values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
        continue
    end

    fX1_cm = fX1/conversion;  fY1_cm = fY1/conversion;
    manOaut = 1; %used auto
    reason = 0; %0 means auto used

    %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
    values(i,:) =
[SUBJ,frame,double(fX1),double(fY1),double(fX1_cm),double(fY1_cm),double(pixI
nt),manOaut,reason];
    save('values.mat','values')   %save the matrix as matlab file
    %call function to plot manual and auto
    fxnplotA(plotX1, plotY1, plotX2, plotY2, fX1, fY1, i, img, PLANE)



    %%%[subj,frame#,autoXpix,autoYpix,autoXcm,autoYcm,pix_int,
manOaut,reason];
    V = array2table(values,'VariableNames',{'SUBJ' 'Frame' 'autoX_pix'
'autoY_pix' 'autoX_cm' 'autoY_cm' 'PixIntensity' 'manOaut' 'reason'});
    %save to excel every 50 frames
    if i == 50
        writetable(V,writefile)
    elseif mod(i,50) == 0
        writetable(V,writefile)
    elseif i == framestop
        writetable(V,writefile)
    end

    %print out what subject & frame we are on
    disp(['Processed subject ', int2str(SUBJ), ', frame ',int2str(frame)])

end  %end of auto loop for frames start - stop
```

```
V = array2table(values,'VariableNames',{'SUBJ' 'Frame' 'autoX_pix'
'autoY_pix' 'autoX_cm' 'autoY_cm' 'PixIntensity' 'manOaut' 'reason'});
writetable(V,writefile)



elapsedTime = toc



end %end of function
```

BIBLIOGRAPHY

1. Stevenson, J.H. and T. Trojian, *Evaluation of Shoulder Pain.* J Fam Pract, 2002. **51**(7): p. 605-611.

2. Lawrence, R.L., et al., *Comparison of 3-dimensional shoulder complex kinematics in individuals with and without shoulder pain, part 1: sternoclavicular, acromioclavicular, and scapulothoracic joints.* J Orthop Sports Phys Ther, 2014. **44**(9): p. 636-45, A1-8.

3. Murphy, R. and A. Carr, *Shoulder Pain.* Am Fam Physician, 2011. **83**(2): p. 137-138.

4. Ludewig, P.M. and J.F. Reynolds, *The Association of Scapular Kinematics and Glenohumeral Joint Pathologies.* J Orthop Sports Phys Ther, 2009. **39**(2): p. 90-104.

5. Hagberg, M. and D. Wegman, *Prevalence rates and odds ratios of shoulder-neck diseases in different occupational groups.* BR J Ind Med, 1987. **44**: p. 602-610.

6. Lo, Y., Y. HSU, and K. Chan, *Epidemiology of shoulder impingement in upper arm sports events.* Br J Sports Med, 1990. **24**: p. 173-177.

7. Fleisig, G.S., et al., *Kinematic and kinetic comparison of baseball pitching among various levels of development.* J Biomech, 1999. **32**: p. 1371-1375.

8. Bayley, J., T. Cochran, and C. Sledge, *The weight-bearing shoulder. The impingement syndrome in paraplegics.* Journal of Bone and Joint Surgery A, 1987. **69**(5): p. 676-678.

9. van der Windt, D., et al., *Shoulder disorders in general practice: prognostic indicators of outcome.* Br J Gen Pract, 1996. **46**(410): p. 519-523.

10. Paine, R. and M. Voight, *The role of the scapula.* J Orthop Sports Phys Ther, 1993. **18**: p. 386-391.

11. Paine, R. and M.L. Voight, *The Role of the Scapula.* The International Journal of Sports Physical Therapy, 2013. **8**(5): p. 617-629.

12. Sheikhzadeh, A., et al., *Three-dimensional motion of the scapula and shoulder during activities of daily living.* J Shoulder Elbow Surg, 2008. **17**(6): p. 936-42.

13. Kibler, W.B. and A. Sciascia, *Current concepts: scapular dyskinesis.* Br J Sports Med, 2009. **44**(5): p. 300-5.

14. Kibler, B.W. and J. McMullen, *Scapular Dyskinesis and Its Relation to Shoulder Pain.* J Am Acad Orthop Surg, 2003. **11**(2): p. 142-151.

15. Kibler, B.W., et al., *Clinical Implications of scapular Dyskinesis in Shoulder Injury.* Br J Sports Med, 2013. **47**(14): p. 877-885.

16. McClure, P., et al., *A Clinical Method for Identifying Scapular Dyskinesis, Part 1: Reliability.* Journal of Athletic Training, 2009. **44**(2): p. 160-164.

17. Tate, A.R., et al., *A Clinical Method for Identifying Scapular Dyskinesis, Part 2: Validity.* Journal of Athletic Training, 2009. **44**(2): p. 165-173.

18. Uhl, T.L., et al., *Evaluation of clinical assessment methods for scapular dyskinesis.* Arthroscopy, 2009. **25**(11): p. 1240-8.

19. van Andel, C., et al., *Recording scapular motion using an acromion marker cluster.* Gait Posture, 2009. **29**(1): p. 123-8.

20. Worobey, L.A., et al., *Reliability of freehand three-dimensional ultrasound to measure scapular rotations.* J Rehabil Res Dev, 2014. **51**(6): p. 985-94.

21. Yu, S., et al., *Automatic identification of needle insertion site in epidural anesthesia with a cascading classifier.* Ultrasound Med Biol, 2014. **40**(9): p. 1980-90.

22. Sonka, M., W. Liang, and R.M. Lauer, *Automated Analysis of Brachial Ultrasound Image Sequences: Early Detection of Cardiovascular Disease via Surrogates of Endothelial Function.* IEEE Transactions on Medical Imaging, 2002. **21**(10): p. 1271-1279.

23. Booth, B. and X. Li, *Boundary Point Detection for Ultrasound Image Segmentation Using Gumbel Disrtibutions* Proc. of Signal Processing for Multimedia Applications, 2007: p. 197-201.

24. Kass, M., A. Witkin, and D. Terzopoulos, *Snakes: Active contour models.* Proceedings of the IEEE International Conference on Computer Vision, 1987: p. 259-268.

25. Itti, L. and C. Koch, *A saliency-based search mechanism for overt and covert shifts of visual attention.* Vision Research, 2000. **40**: p. 1489-1506.

26. Middleton, W.D., A.B. Kurtz, and B.S. Hertzberg, in *Ultrasound, The Requisites*. 2004, Mosby, Inc. p. 3-26.

27. Chai, H.Y., L.K. Wee, and E. Supriyanto. *Edge detection in ultrasound images using speckle reducing anisotropic diffusion in canny edge detector framework*. in *World Scientific and Engineering Academy and Society (WSEAS)*. 2011. Stevens Point, Wisconsin.

28. Shapiro, L.G. and G.C. Stockman, *Computer Vision*. 2001, New Jersey: Prentice-Hall. 279-325.

29. Chen, L., et al., *A quality-guided displacement tracking algorithm for ultrasonic elasticity imaging.* Med Image Anal, 2009. **13**(2): p. 286-96.

30. Solberg, O.V., et al., *Freehand 3D ultrasound reconstruction algorithms--a review.* Ultrasound Med Biol, 2007. **33**(7): p. 991-1009.

31. Wu, G., et al., *ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - Part II: shoulder, elbow, wrist and hand.* J Biomech, 2005. **38**: p. 981-992.

32. Shrout, P. and F. Fleiss, *Intraclass correlations: uses in assessing rater reliability.* Psychol Bull, 1979. **86**(2): p. 420-428.

33. Landis, R.J. and G.G. Koch, *The Measurement of Observer Agreement for Categorical Data*. Biometrics, 1977. **33**(1): p. 159-174.

34. Czerwinski, R.N., D.L. Jones, and J. O'Brien, William D, *Detection of Lines and Boundaries in Speckle Images—Application to Medical Ultrasound*. IEEE Transactions on Medical Imaging, 1999. **18**(2): p. 126-136.

35. Hacihaliloglu, I., et al., *Automatic Data-Driven Parameterization for Phase-Based Bone Localization in US Using Log-Gabor Filters*, in *Advances in Visual Computing*. 2009, Springer Berlin Heidelberg. p. 944-954.

36. Dalvi, R., I. Hacihaliloglu, and R. Abugharbieh. *3D Ultrasound Volume Stitching Using Phase Symmetry and Harris Corner Detection for Orthopaedic Applications*. in *SPIE*. 2010.