© 2018 Peng Li

AN AUTOMATED TOOL FOR MEASURING HUMAN LIMB BONES

BY

PENG LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Associate Professor Derek W. Hoiem

ABSTRACT

Sorting commingled human skeletal remains is one of the most time-consuming jobs for forensic anthropologists. Multiple measurements must be taken from each bone by hand. 3D scanners are an unfeasible alternative because they are expensive and difficult to transport. This thesis presents a software tool that automatically takes multiple measurements on bones using 2D images. Results show that this method takes significantly less time and performs as accurately as standard measurement methods in the field of anthropology. This technology has implications for streamlining data collection practices and, by extension, forensic analyses, and procedures.

To my parents and friends, for their love and support.

ACKNOWLEDGMENTS

I would like to first thank my thesis advisor Professor Derek Hoiem for his constant support, patient guidance and continuous encouragement. This thesis would not have been possible without him.

I would also like to thank Amanda Lee, my friend and research partner, for her valuable and tremendous inputs into this project. She was always ready to answer my questions about forensic science and anthropology.

And finally, I'd like to thank the University of Illinois, which provided research equipment and supported my tuition during my graduate study.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

ABBREVIATIONS viii

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: IMAGE CAPTURE AND ANALYSIS 3

CHAPTER 3: RESULTS AND EVALUATION 17

CHAPTER 4: DISCUSSION AND CONCLUSION 21

REFERENCES 23

APPENDIX A: MEASUREMENTS AND DEFINITIONS 25

APPENDIX B: CHECKERBOARD SAMPLE 26

APPENDIX C: LIST OF CODE FILES 27

LIST OF TABLES

2.1 Descriptive statistics for SEI by variable 18

2.2 Paired T-test values and descriptive statistics for each variable comparing the raw measurements (mm) taken by the software and those taken by hand 20

A.1 Measurements and definitions used in study.....25

LIST OF FIGURES

2.1 System Structure Overview 3

2.2 Pixel distribution for an input image 5

2.3 Final image for count number of radii and original image with bounding box 6

2.4 The output of the bone mask on the input image 8

2.5 System Structure for Bone Segmentation 9

2.6 Original Input Femur Mask vs Rotated Femur 10

2.7 Femur POIs (Red Circles) 11

2.8 Femur after rotation and flipping 12

2.9 Humerus POIs (Red Circles) 13

2.10 Radius POIs (Red Circles) 14

2.11 Tibia POIs (Red Circles) 15

3.1 Box plot of SEI sorted by variable and grouped by bone 19

3.2 Box plot of absolute error in millimeters sorted by variable and grouped by bone 19

B.1 Sample of the Checkerboard Pattern 26

ABBREVIATIONS

ICC	Intraclass correlation coefficient
FBML	Femur Bicondylar Length
FEB	Femur Epidondylar Breadth
FHD	Femur Head Diameter
FML	Femur Maximum Length
FMLD	Femur Mediolateral Diameter
HEB	Humerus Epicondylar Breadth
HHD	Humerus Head Diameter
HML	Humerus Maximum Length
HSL	Hue, Saturation, Lightness
POI	Point of Interest
RML	Radius Maximum Length
RMLD	Radius Mediolateral Diameter
ROI	Region of Interest
SEI	Scaled Error Index
TML	Tibia Maximum Length
TPB	Tibia Plateau Mediolateral Breadth

CHAPTER 1

INTRODUCTION

The commingling—mixing—of human skeletal remains has always presented a series of problems for forensic anthropologists [1]–[3]. In order to identify and repatriate exhumed remains to their relatives, it is necessary to first sort the bones and assign them to single individuals. Standard sorting methods use algorithms, which require multiple linear measurements to be taken from each bone. Currently, these measurements are taken by hand using spreading and sliding calipers, a mandibulometer, metal tape, and an osteometric board [4]. This process is time-consuming, and the error rates are highly dependent on the experience level of the specialist [5]. Another disadvantage of this method is that the measurement instruments are hard to carry to the field, which is required in most contexts of commingled remains (e.g., sites of mass graves, natural disasters, and terrorist attacks). Laser scanners are sometimes used to measure bones in research settings, but they are expensive—from thousands of dollars to tens thousands of dollars, difficult to transport and are slower than the process of measuring by hand.

To address these issues, we propose a new measurement tool that automatically analyzes 2D images of long bones. As smartphones are currently the most popular personal electronic devices, we consider using them to take the measurements on bones. Additionally, the continuing development of smartphone cameras allows us to take high resolution images, making accurate measurements possible. Overall, the smartphone is an affordable and accessible replacement for the laser scanner.

The method developed in this project works as follows: The user takes high resolution images of bones with a smartphone, importing them into a software tool. Then the program automatically detects the measurements that need to be taken and outputs the measurements for these bones.

The rest of the thesis is organized as follows:

- Chapter 2 explains the specifications to take photos of the human limb bone, the components for the software tool and the algorithms behind it.

- Chapter 3 shows the final measurement results of the software tool.
- Chapter 4 discusses the software tool and concludes the thesis.

CHAPTER 2

IMAGE CAPTURE AND ANALYSIS

In this chapter, we discuss the image capture and analysis procedures. There are three main steps for the auto-measurement tool. The first step is to collect images of the bone—taking photos of the bones with a predefined background. The second step is to calibrate the camera to calculate the intrinsics of the camera. Finally, the third step is to analyze these images: first, undistorting the image and computing rotation and translation of the camera; second, counting the number of bones inside the image; third, segmenting the bone area; fourth, finding point of interests (POIs); and finally, calculating the distance between different points in real world coordinate system.

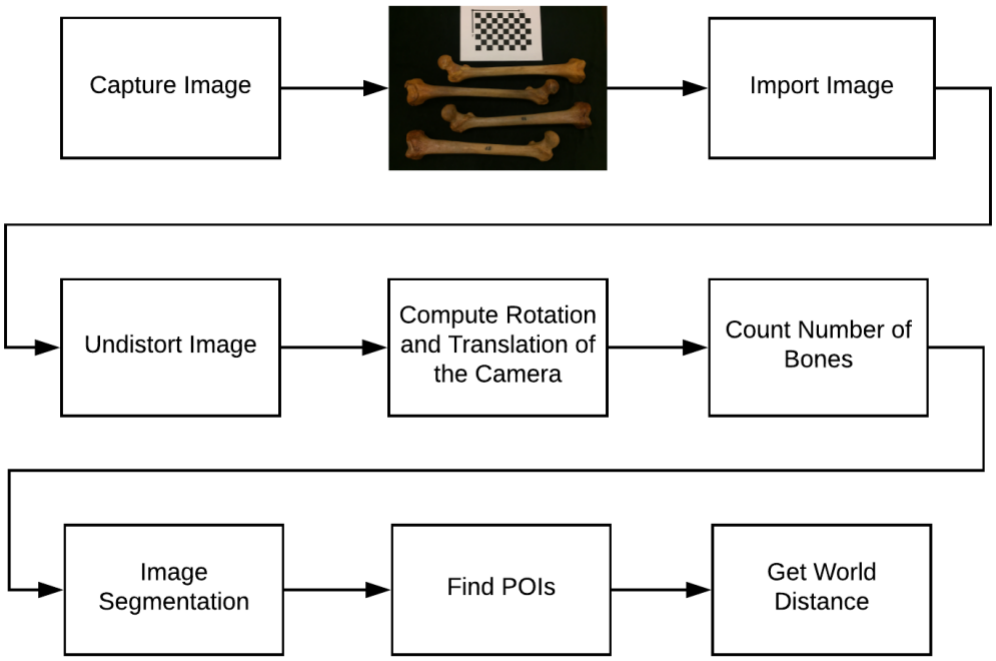


Figure 2.1 System Structure Overview

2.1 Image Capture

To reduce the amount of computing time and to improve the measurement accuracy in the image analysis step, we use a black velvet fabric background for the images. In addition, a black-and-white checkerboard, used as a calibration target, is placed near the bone(s).

Therefore, both the background (black velvet fabric) and the calibration target (checkerboard) only have black-and-white colors. This makes the image segmentation step, which will be discussed below, more accurate.

2.2 Camera Calibration

In order to estimate camera intrinsics, extrinsics, and lens distortion parameters, the camera needs to take around 20 images with the calibration target in different positions. Then the images are imported into an camera calibration function `estimateCameraParameters` in MATLAB® [6] to estimate camera intrinsics, extrinsics, and lens distortion parameters.

2.3 Image Analysis

In this section, we will discuss the steps to take measurements of four types of human limbs: femur, humerus, tibia, and radius.

2.3.1 Undistort Image

Unlike real world images, the camera images are radial distorted due to the bending of light rays at the edges of the lens. To obtain correct measurements from the images, the images are first undistorted using the function `undistortImage` in MATLAB® [6] to account for radial distortion. The relationship between a distorted point $(x_{distorted}, y_{distorted})$ and an undistorted point (x, y) is shown below:

$$\begin{aligned} x_{distorted} &= x(1 + k_1 \times (x^2 + y^2)^2 + k_2 \times (x^2 + y^2)^4) \\ y_{distorted} &= y(1 + k_1 \times (x^2 + y^2)^2 + k_2 \times (x^2 + y^2)^4) \end{aligned} \tag{2.1}$$

where k_1 and k_2 are radial distortion coefficients of the lens.

2.3.2 Compute Rotation and Translation of the Camera

The rotation and translation matrixes represent the position of the camera. A MATLAB® function, `extrinsics` [6], is called to calculate the rotation and translation. The function has three inputs: the world coordinates and the corresponding image coordinates, which are both derived from the checkerboard, and the camera parameters, which are estimated from step 2.2.

2.3.3 Count Number of Bones

To count the number of bones in the image, the first step is to segment the bones from the background. Then we count the number of bones by counting the number of large blobs.

Since the input image is taken with a black-white checkerboard and a black velvet fabric background, the regions of interest contain more pixels with color than the monochrome pixels. Thus, the image can be interpreted with Hue, Saturation, Lightness (HSL) representation. Specifically, we utilize saturation and lightness of the image with Otsu's method [7] to find large blobs of bone region.

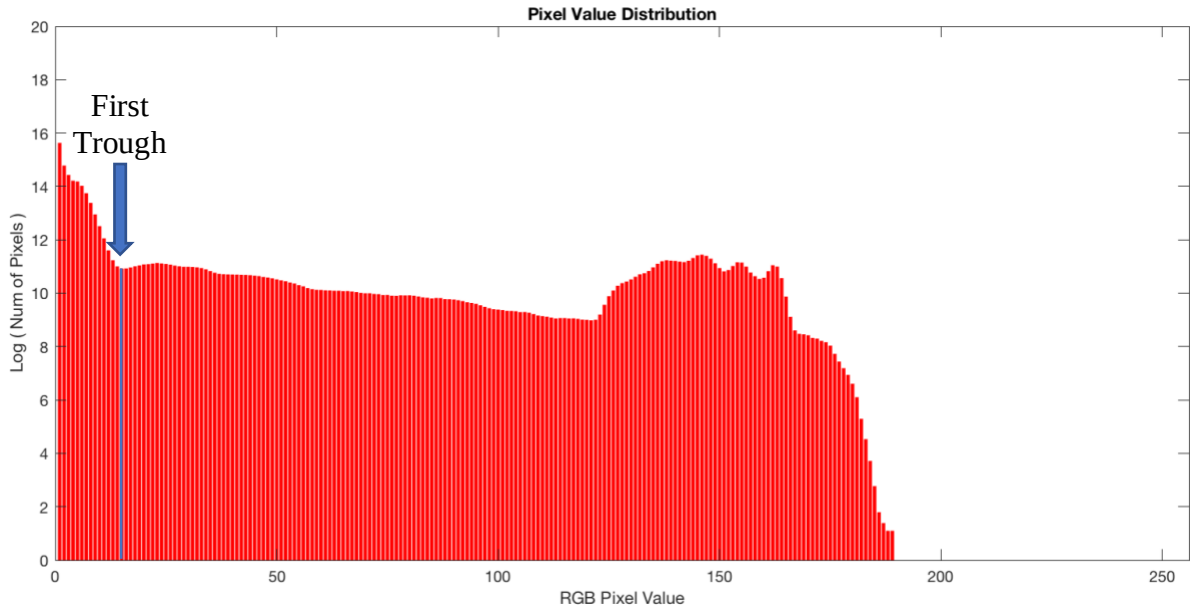


Figure 2.2 Pixel distribution for an input image (blue line is the first trough)

Although the picture of the bone(s) is taken with a black-white background, depending on the background light and camera settings, the background is not always completely black (i.e. it has the relatively same saturation value as pixels of the bone). To solve this problem, a force-black filter is applied to the image first to make 70% of the "dark pixels" totally black (RGB = [0, 0, 0]). "Dark pixels" are defined as pixels with R, G, B values that are all smaller than a certain value. We select the pixels with R, G, B values that are all smaller than 70% of the first trough value in the distribution, shown in Figure 2.2.

After filtering, the image is represented with HSL format. In the lightness channel, the bone, the checkerboard and some noise are obvious. In the saturation channel, some noise on the edge of the image is always visible. Depending on the color of the bone, the noise is more conspicuous (e.g., if the bone is bleached, the noise has high saturation value).

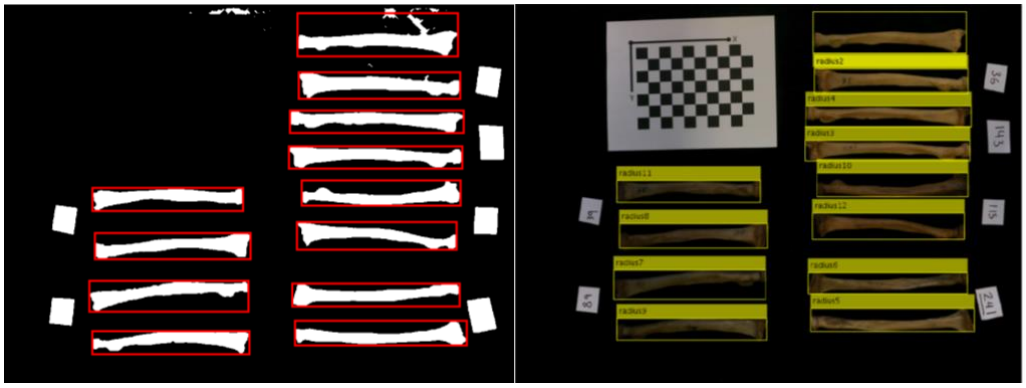


Figure 2.3 Final image for count number of radii and original image with bounding boxes

To find the general area of the bone, we first remove the checkerboard from the image using the lightness channel. Compared with the bone region, the checkerboard region has a similar height and width. Thus, we can easily identify the checkerboard area by limiting the number of pixels in a section and the height, width value. A black (RGB = [0, 0, 0]) value is assigned to all of the checkerboard pixels. Next, the lightness and saturation information are extracted from the checkerboard-removed image. After that, the lightness and saturation intensity images are converted to a binary image with a threshold determined by Otsu's method [7]. These two images are combined together using logical disjunction between corresponding pixels. The noise is removed by active contours algorithm [8]–[10]. The active contours algorithm minimizes energy

along a spline structure, where the noise pixels can be easily removed. As a result, large regions of interest can easily be identified and counted.

At the end of this function, the number of bones is outputted along with a bounding box for each large region of interest (ROI). Iteratively, the regions of interest will be processed by the next three steps: image segmentation, find POIs, and get world distance.

2.3.4 Image Segmentation

Automatic image segmentation is used to separate the ROI from the background with predefined criteria. It is one of the more difficult challenges in the field of computer vision because images have varied inputs and the predefined criteria might not apply to all cases. The image segmentation function is one of the most important components in this system. If the ROI is not completely segmented out, the measurements will be shorter than the actual length. On the other hand, if the ROI is segmented out with noise on the edges, the measurements will be larger than the actual length. Thus, the bone region must be segmented out as precisely as possible.

Since long bones have an uneven surface, the edges of the long bones in the image are very dark and have nearly the same saturation value as the background. Conversely, the lightness of the bone region is highly contrasted with the background. In our algorithm, the saturation channel is firstly binarized by the Otsu's threshold method. Then the sets of noise pixels, which cannot be reached by filling in the background from the edge of image, are removed using MATLAB® function `imfill` [6]. The largest region is picked out and combined with the lightness channel using logical disjunction.

A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image. To get the initial bone mask, the holes are removed from the combined image first. Next, to make sure the bone area is totally covered by the mask, the image is dilated using a small circle ($diameter = 10\ pixels$). Then we apply the active contour algorithm [8]–[10] to find the edges of the ROI. Because the edge of the bone is darker, which makes the edge pixels more similar to the background pixels, the active contour algorithm always finds a smaller bone region than the actual bone region if we apply around 35 iterations in the algorithm. One way to

reduce such error could be reduce the number of iterations in the algorithm. However, our initial runs show that reducing the number of iterations result in jagged-edge of the bone and the attachment of tiny blobs to the bone. To solve the over-segmentation problem, the smaller bone region obtained by active contour algorithm is applied to the input image as a black-mask. Then the dim edges become much more visible in the saturation channel as almost all the ROI has been masked with black pixels. Again, the image is binarized with Otsu's threshold. The bone edge region is combined with the smaller bone region to obtain the full region of bone. Finally, to smooth the edges of the region, a Gaussian filter is applied to the image, and then the image is binarized using Otsu's threshold.

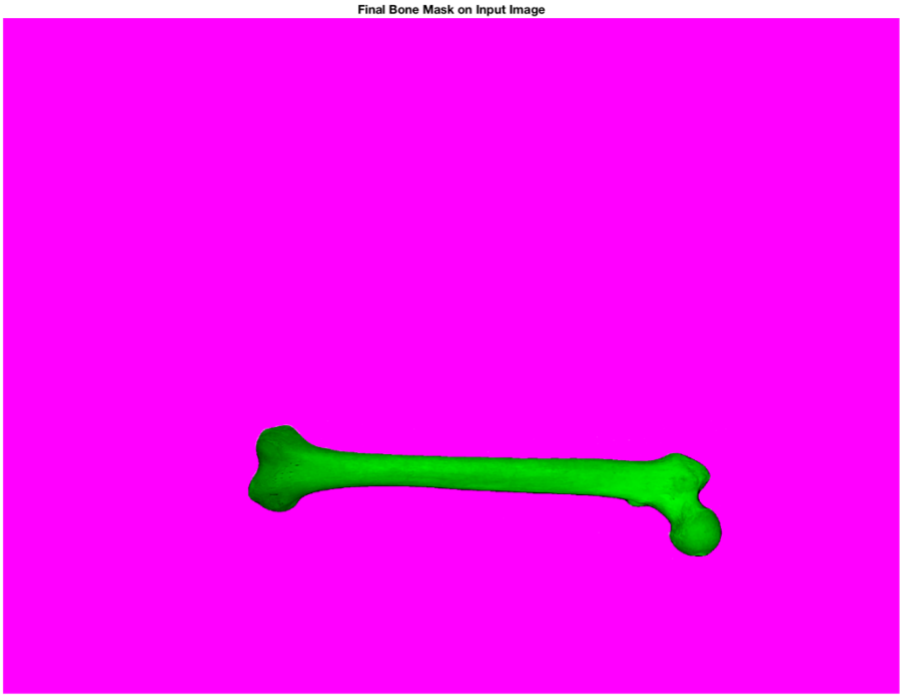


Figure 2.4 The output of the bone mask (purple) on the input image (green)

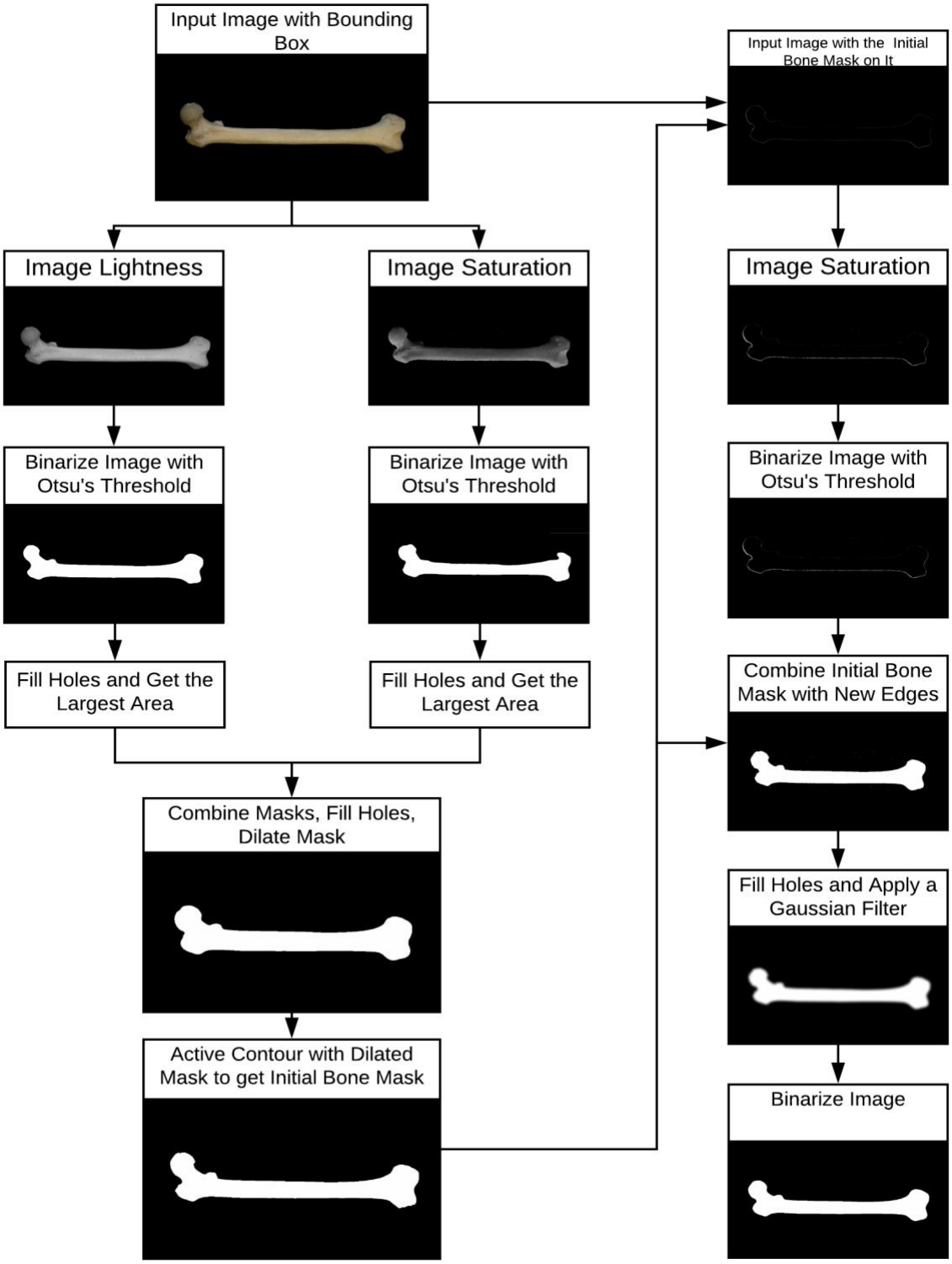


Figure 2.5 System Structure for Bone Segmentation (the images are zoomed in to show more details)

2.3.5 Find POIs

To measure the length, we need find POIs (Point of Interests) for different measurements on the binary bone images. Though some of the measurement methods between different bone types are the same, other measurement methods are special for each type of bones. All bones must be rotated to a horizontal position, in order to measure the maximum length using a bounding box. We have developed different methods for each bone to make sure they all have the same layout before the software finds the POIs. We explain the bone rotation process below. The unique methods for each type of bone are introduced in the following subsections.

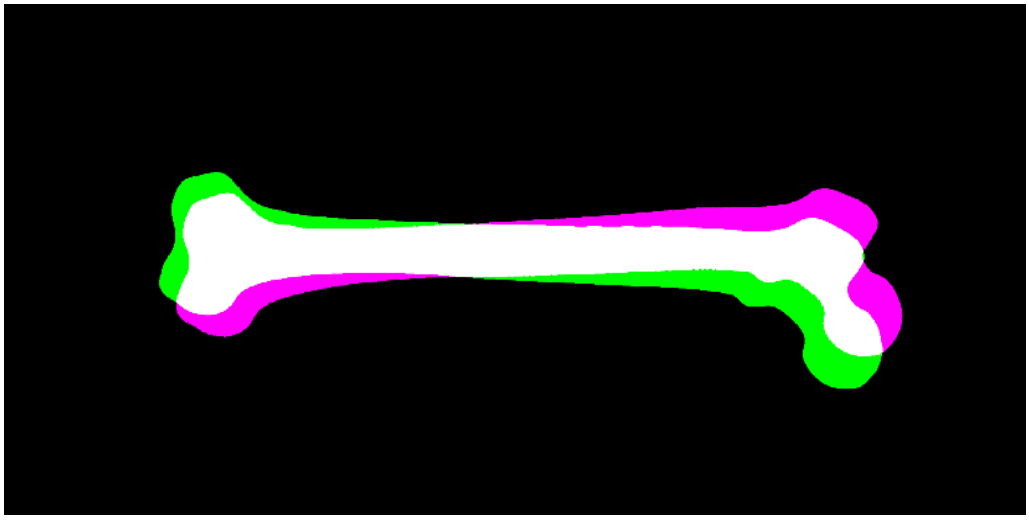


Figure 2.6 Original Input Femur Mask (Green & White) vs Rotated Femur (Purple & White)

To rotate the bone region, an ellipse, which has the same second-moments as the ROI, is constructed. Then the ROI is rotated for an angle between the horizontal axis and major axis of the ellipse. Then we can easily find a bounding box on the bone region by finding the farthest left, right, top and bottom pixels.

2.3.5.1 Find POIs for Femur

The femur is the largest bone in the human body. Five measurements are taken for it: FBML, FEB, FHD, FML, and FMLD.

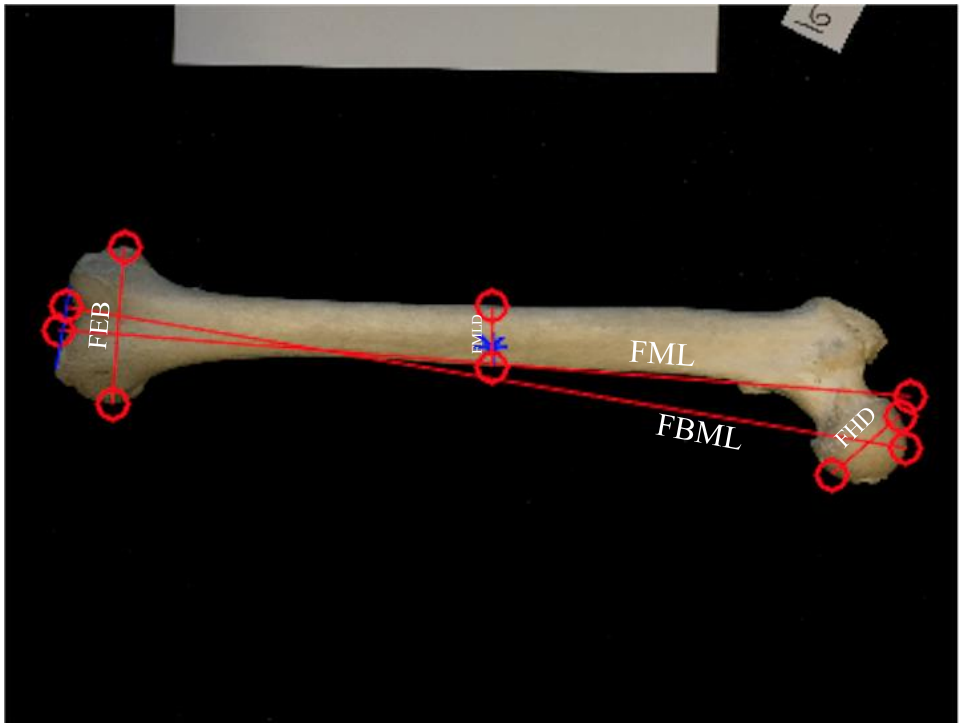


Figure 2.7 Femur POIs (Red Circles)

In order to find the accurate POIs for femur the ROI must be flipped so that every femur is in the same position before the algorithm looks for the POIs. In our software, we always place the head of the femur on the right bottom corner of the image. After observing and testing multiple femurs, we found that the centroid (blue point in Figure 2.8) of the middle shaft ROI is always in the top-half of the region. So, the image is flipped along the Y axis if the centroid is on the bottom-half of the region in order to make sure that the head of the femur is facing downward.

Next, we need to distinguish the head of the femur from the other, distal, end. In order to do this, the bone is divided into ten regions along the X axis. We then apply two new bounding boxes to one section on either end (i.e., 1/10th of the region on either side), shown in Figure 2.8. The bounding boxes are cropped to the height of the bone section, and the percentage of background pixels within each respective bounding box is found. The head region always has a larger percentage of background pixels within the bounding box, due to its irregular shape. As a result, the software is able to identify the head of the femur and if it is on the left side, the image is flipped along the X axis.

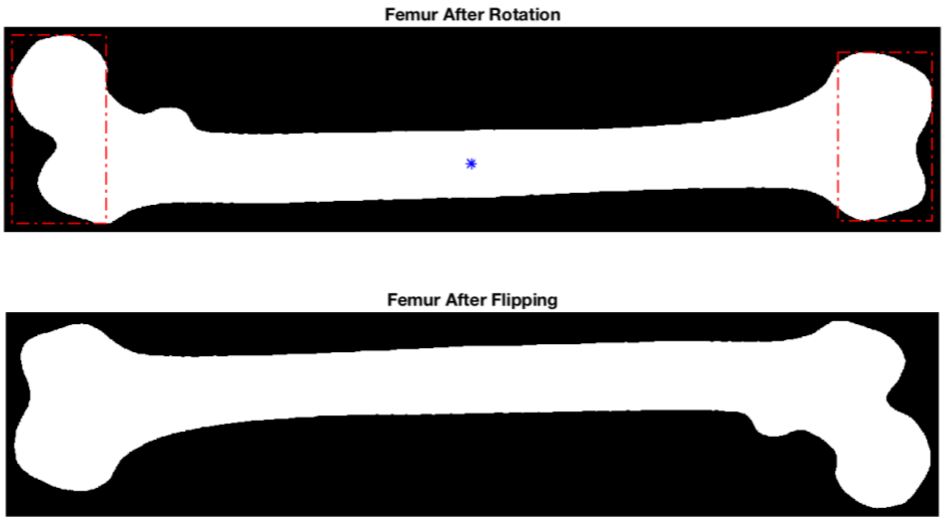


Figure 2.8 Femur after rotation and flipping (Red boxes are the bounding boxes, and blue point is the centroid)

To measure FHD, a line segment scans the head of the region with an angle of 45 degrees (the angle between the line segment and the horizontal axis). The scan stops when the line segment reaches the highest value.

To measure FMLD, we first find the centroid of the bone region. Then shortest length of a line segment is identified where the length the distance between the bone edge on the top of the image and bottom of the image.

To measure FEB, the left one third of the bone region is segmented out. FEB POIs are identified as the intersect point on the top of the image with the bounding box and the corresponding bottom point on the bounding box with the same column.

To measure FBML, first we find a middle peak of a horizontal length in the left one-third of the bone region, where the horizontal length is the column number of the first pixel of the bone region counting from the left. Next, two minimum values are identified on each side of the peak. Then, two points are identified and can be connected using a line L. The FBML is the maximum length from a point on the edge of the region to the line L. This measurement is easily obtained by iterating all edge points on the region.

2.3.5.2 Find POIs for Humerus

The humerus is a midsize bone of the human limb bones. Three measurements have been taken for it: HML, HEB, and HHD.



Figure 2.9 Humerus POIs (Red Circles)

The primary challenge for the humerus is to identify the head region of the humerus. In our software, we always put the head on the right bottom corner of the image. To identify the head region, our algorithm finds circles using the circular Hough transformation [11]–[13]. An adaptive sensitivity is developed in our algorithm (e.g., sensitivity for finding a circle is set to a relatively small value and increased by a step value if no circle is detected). After, the center of the head is found. Next, two points that lie on a line 45 degrees to the line segment, which crosses the center of the head, are identified by finding the crossing points of the line segment and the edge of the bone region. After that, these two points are moved along the edges of the bone region to find the minimum distance, which makes one point fall within the concave of the shape. Then we move the point down to find the correct maximum diameter measurement.

The process of measuring HEB is the same as FEB. Please refer to FEB for the method.

2.3.5.3 Find POIs for Radius

The radius is the smallest bone of the human limb bones. Two measurements have been taken for it: RML and RMLD. Since the measurement for RML is the same as FML and the measurement for RMLD is the same as FMLD, please refer FML and FMLD for these measurement methods.

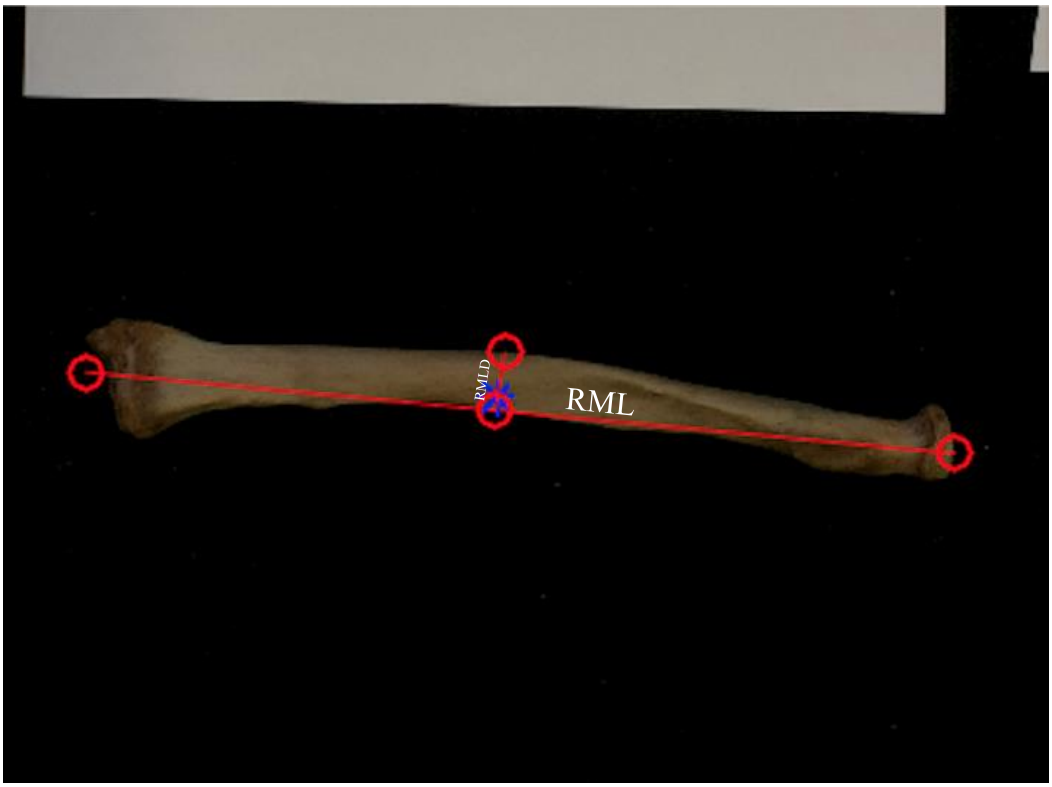


Figure 2.10 Radius POIs (Red Circles)

2.3.5.4 Find POIs for Tibia

The tibia is the second largest bone of the human limb bones. Two measurements have been taken for it: TML and TPB. The process for measuring TML, is similar to the one for FML. The only difference is that on the proximal side, a lower top is used for one of the vertexes. To measure TPB, we segment out the proximal part of the bone. Then the most medial and lateral points are identified with the bounding box on the sub-region. These two points are the two vertexes for TPB.

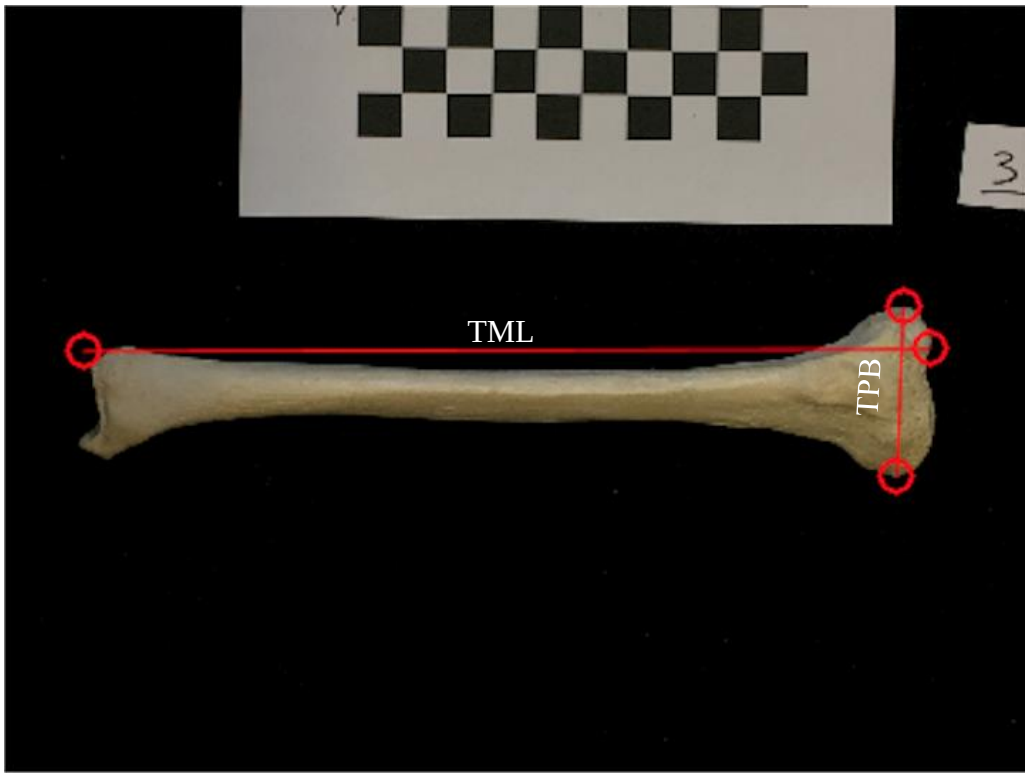


Figure 2.11 Tibia POIs (Red Circles)

2.3.6 Get World Distance

Finally, all the POIs that are found in the previous section are converted back to the world coordinates using the camera model [14]. The camera model is defined as:

$$x = K\ [R\ t]\ W \tag{2.2}$$

where **x** is the image coordinates $(u, v, 1)$, **K** is a 3 by 3 intrinsic matrix, **R** is a 3 by 3 rotation matrix, **t** is a 3 by 1 translation matrix and **W** is world coordinates $(X, Y, H, 1)$. Here, the POIs from previous sections are image coordinates; **K**, **R**, and **t** has been solved in the previous sections; and the height (H) of the point is estimated by an algorithm described below.

We estimated the height of POIs from the surface of the table to reduce the software measurement error of bones. H is a necessary parameter because it affects the distance between the POIs and the camera. If the POIs are closer to the camera (i.e., the height of the bone is larger), then the

algorithm will give a larger measurement estimate than if they are farther away. By estimating H we account for this error.

In order to estimate H, first, we take pictures of 30 bones with the actual measurements already recorded. Second, the software measurements are computed with the different estimated heights. Then we use the H that creates the least software measurement error. The software measurement error is defined as follows:

$$Error = \frac{\sum_{Number\ of\ Bones}(Measurement\ Length - Actual\ Length)^2}{Number\ of\ Bones} \tag{2.3}$$

Pseudo code for finding height:

1. Initialize $H_{low} = 0, H_{high} = 100, Step = 10$;
2. Initialize $Delta = Infinity, Error_{current} = Infinity$;
3. While $Delta$ > 0.01:
4. $Error_{previous} = Error_{current}$;
5. For $H = H_{low}: Step: H_{high}$:
6. Find $Measurement\ Length$ by solving $x = K\ [R\ t]\ W$;
7. Find the $Error_{current}$ using the Error Equation (2);
8. Find **Minimum** $Error_{current}$ and corresponding height H';
9. $H_{low} = H' - Step, H_{high} = H' + Step$;
10. $Step = Step/5$;
11. $Delta = Abs(\ Error_{current} - Error_{previous})$;
12. Return $H_{mean} = (H_{high} + H_{low})/2$.

CHAPTER 3

RESULTS AND EVALUATION

In this chapter, we discuss the measurement error data and the feasibility of this software as a viable measurement tool. A total of 12 postcranial measurements (Appendix A) were taken from a sample of 37 individual human skeletons housed at the University of Illinois at Urbana-Champaign, Department of Anthropology [4]. Some measurements were not recorded because they were compromised by physical damage or pathological features of the bone (e.g., osteoarthritis, periostitis, etc.). Measurements for each bone were chosen based on their discriminant value for sorting commingled remains, and their ability to be measured in a single plane. The data were "cleaned" prior to statistical analyses by removing outliers that were beyond the range of possible human variation.

In order to test the accuracy of the software we used the Scaled Error Index (SEI) [5]:

$$Scaled\ Error\ Index = \frac{|Raw\ Measurement - True\ Median|}{True\ Median} \times 100 \tag{3.1}$$

The raw measurement represents the measurement recorded by the algorithm. In order to obtain the raw measurement, it is necessary to first estimate the height of the bone so as to eliminate measurement error. The height is estimated using a Leave-One-Out (LOOCV) procedure. LOOCV removes a single observation from the original dataset as the test data, processes the remaining individuals using the training data, and then repeats over again using another individual. The predictions then represent the raw measurements. Once the height of a single bone is predicted using LOOCV, the rest of the (raw) measurements of the single bone are calculated based on the predicted height. The true median was derived by taking the median of three separate repeated "by-hand" measurements for each variable using an osteometric board and digital calipers, to the millimeter. This allows us to compare the software's results to the best measurement recordable using standard anthropological methods. Furthermore, this index can be compared between bones, regardless of scale. This is an important feature considering the variable size of limb bones.

Using the SEI we are testing the *validity*—used interchangeably in this paper with *accuracy*—of the software. This is defined as the degree to which a measurement measures what it is purported to measure [15]. Inaccurate measurements can lead to erroneous estimations of biological details, which can interfere with, or even potentially, derail forensic investigations [16]. Taking this into consideration, it is critical to evaluate this aspect of the method.

By far, the measurement with the largest SEI (lowest accuracy) was RMLD (Table 3.1). The measurement with the next largest SEI was TPB. In both Figure 3.1 and Figure 3.2, the measurements for FHD, HEB, RMLD have relative high error. The high error rate is likely due to the significant curvatures of the bones at these landmarks. There is more room for variability because the recorded measurements are highly dependent on the angle of the camera. All variables containing length had relatively low SEIs, suggesting that these are relatively robust measurements.

Table 3.1 Descriptive statistics for SEI by variable

Measurement	*n*	Mean	Standard Deviation
FBML	18	0.626	0.284
FEB	17	1.500	1.510
FHD	18	2.650	1.890
FML	18	0.537	0.397
FMLD	18	1.190	0.816
HEB	24	1.790	1.770
HHD	23	2.570	2.390
HML	24	0.261	0.199
RML	17	0.460	0.413
RMLD	19	6.250	6.180
TML	19	0.491	0.374
TPB	17	2.700	2.490

The measurements with the highest accuracy also had the highest precision. This could be due to the variability of the camera angle or the discrepancies between the interpretation of the measurement description by the algorithm and the osteologist recording by hand. In the case of the

latter, the algorithm can be adjusted to account for these inconsistencies in order to increase accuracy and precision rates.

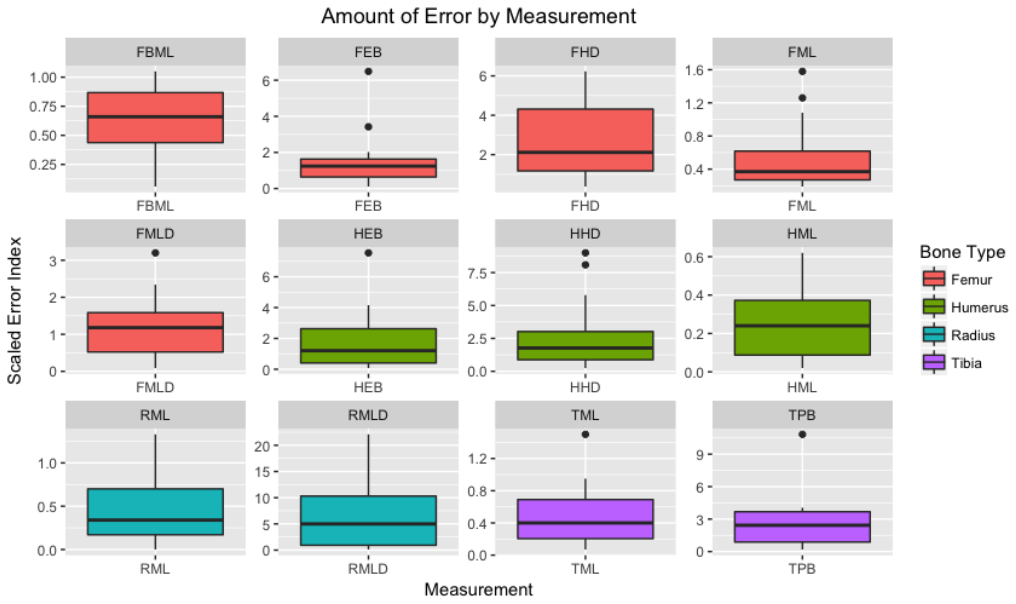


Figure 3.1 Box plot of SEI sorted by variable and grouped by bone

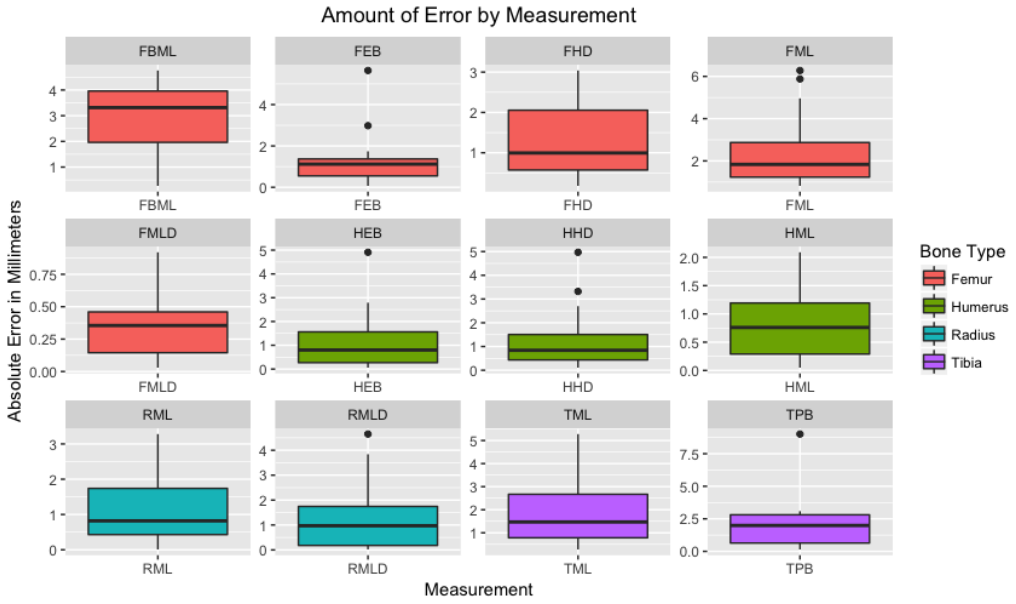


Figure 3.2 Box plot of absolute error in millimeters sorted by variable and grouped by bone

In order to determine whether the differences between measurements done by hand and those taken by the software were significant, paired t-test comparisons were performed. Shapiro-Wilk tests showed that the differences between measurements were normally distributed, satisfying the

assumption of normality for t-tests. The *p*-values for each variable in Table 3.2 were all above 0.87. These results indicate that the two methods produce similar measurements.

Table 3.2 Paired T-test values and descriptive statistics for each variable comparing the raw measurements (mm) taken by the software and those taken by hand

		Software		By hand		
Measurement	*n*	Mean	Standard Deviation	Mean	Standard Deviation	*p*
FBML	18	459.00	32.10	460.00	31.70	0.96
FEB	17	84.10	7.22	84.10	6.97	0.94
FHD	18	48.50	4.02	48.60	3.80	0.92
FML	18	463.00	33.00	463.00	32.00	0.92
FMLD	18	29.10	3.06	29.10	2.94	0.90
HEB	24	65.50	4.11	65.50	4.30	0.99
HHD	23	48.80	2.81	48.80	3.44	0.98
HML	24	329.00	19.40	329.00	18.70	0.87
RML	17	243.00	11.50	243.00	11.60	0.99
RMLD	19	17.90	1.66	18.00	1.94	0.92
TML	19	384.00	19.20	384.00	18.30	0.94
TPB	17	77.90	4.42	78.00	3.75	0.92

We also evaluate the results using intraclass correlation coefficient (ICC). The ICC is used to access the consistency of measurements made by multiple observers measuring the same quantity [17]. The range for ICC is from 0 to 1. The higher ICC is, the more consistent the measurement is. Our results show that the ICC value between the measurements done by software and measurements done by hand is 0.999881, which is an excellent ICC inter-rater agreement value [18].

CHAPTER 4

DISCUSSION AND CONCLUSION

In this thesis, we presented a software tool that automatically measures multiple sets of data for bones using 2D images as input. The software is able to measure multiple bones of the same type in a single image. A high precision image segmentation method is proposed and implemented in the software. Unique algorithms are developed for different type of bones. The software tool significantly reduces the amount of time for obtaining bone measurements and reaches the same level of accuracy as the standard measurement methods in the field of anthropology.

Implementation of this software in standard anthropological practices has implications for streamlining medicolegal investigations, improving the quality of anthropological research, and increasing the speed of site assessment. Before the creation of automated analyses, evaluations of large-scale commingled assemblages were infeasible due to the time-consuming nature of non-metric estimation techniques. Now, there are algorithms for the matching of bones [1], [19]–[21], but there is still no practical technology for the automated recordation of bone measurements in the field. Bone measurements are necessary input for the matching algorithms, and currently they are all taken by hand. This software has the potential to drastically reduce the amount of time, energy, and manpower needed to conduct skeletal sorting procedures.

The future, effective implementation of this algorithm was a primary consideration throughout the development of this project. The ubiquity of the smartphone ensures that nearly all specialists in the field will be able to access and use this software. In addition, it is standard practice for osteologists to carry black fabric with them in order to take pictures of skeletal material recovered from the field. With this in mind, the algorithm was designed to work using a black background. The only necessary implement for this process, which an anthropologist is not already likely to be carrying, will be the checkerboard scale. However, the scale can easily be printed out.

Future directions of this work include the development of a smartphone application integrating this software that can take photos, calibrate the camera and measure human limbs. Additionally, a

stereo camera could be used to take photos, from which it is possible to extract 3D information of the bones.

REFERENCES

[1] J. E. Byrd and B. J. Adams, “Osteometric Sorting of Commingled Human Remains,” *J. Forensic Sci.*, vol. 48, no. 4, pp. 717–724, 2003.

[2] N. P. Herrmann, J. B. Devlin, and J. C. Stanton, *Commingled Human Remains*, no. Adams 1996. Elsevier, 2014.

[3] B. J. Adams and J. E. Byrd, “Resolution of small-scale commingling: A case report from the Vietnam War,” *Forensic Sci. Int.*, vol. 156, no. 1, pp. 63–69, 2006.

[4] J. E. Buikstra and D. H. Ubelaker, “Standards for Data Collection from Human Skeletal Remains,” *Arkansas Archeol. Surv. Res. Ser. No. 44*, no. 4, p. 206, 1994.

[5] B. J. Adams and J. E. Byrd, “Interobserver Variation of Selected Postcranial Skeletal Measurements,” *J. Forensic Sci.*, vol. 47, no. 6, p. 15550J, 2002.

[6] The MathWorks Inc., “Computer Vision System Toolbox.” Natick, Massachusetts, United States, 2017.

[7] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[8] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, 2001.

[9] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic Active Contours,” *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[10] R. T. Whitaker, “A Level-Set Approach to 3D Reconstruction From Range Data,” *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 203–231, 1998.

[11] T. J. Atherton and D. J. Kerbyson, “Size invariant circle detection,” *Image Vis. Comput.*, vol. 17, no. 11, pp. 795–803, 1999.

[12] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, “Comparative study of Hough Transform methods for circle finding,” *Image Vis. Comput.*, vol. 8, no. 1, pp. 71–77, 1990.

[13] E. R. Davies, “Machine Vision, Third Edition: Theory, Algorithms, Practicalities,” *Pattern Recognit. Lett.*, vol. 7010, p. 934, 2004.

[14] J. Ponce and D. Forsyth, *Computer vision: a modern approach*. 2012.

[15] J. D. Nance, “Reliability, validity, and quantitative methods in archaeology,” in

Quantitative research in archaeology, M. S. Aldenderfer, Ed. Newbury Park: Sage Publications, 1987, pp. 245–293.

[16] R. L. Jantz, D. R. Hunt, and L. Meadows, "The measure and mismeasure of the tibia: implications for stature estimation.," *J. Forensic Sci.*, vol. 40, no. 5, pp. 758–761, 1995.

[17] P. E. Shrout and J. L. Fleiss, "Intraclass correlations: Uses in assessing rater reliability," *Psychol. Bull.*, vol. 86, no. 2, pp. 420–428, 1979.

[18] D. V. Cicchetti, "Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology," *Psychol. Assess.*, vol. 6, no. 4, pp. 284–290, 1994.

[19] R. M. Thomas, D. H. Ubelaker, and J. E. Byrd, "Tables for the metric evaluation of pair-matching of human skeletal elements," *J. Forensic Sci.*, vol. 58, no. 4, pp. 952–956, 2013.

[20] J. J. Lynch, J. Byrd, and C. B. Legarde, "The Power of Exclusion using Automated Osteometric Sorting : Pair-Matching † , ‡," *J Forensic Sci*, pp. 1–10, 2017.

[21] J. J. Lynch, J. Byrd, and C. B. Legarde, "The Power of Exclusion using Automated Osteometric Sorting: Pair-Matching," *J. Forensic Sci.*, pp. 1–10, 2017.

APPENDIX A

MEASUREMENTS AND DEFINITIONS

Measurement	**Definition**
Femur Bicondylar Length (FBML)	Distance from the most superior point on the head to a plane drawn along the inferior surfaces of the distal condyles
Femur Epidondylar Breadth (FEB)	Distance between the two most laterally projecting points on the epicondyles
Femur Head Diameter (FHD)	The maximum diameter of the femur head
Femur Maximum Length (FML)	Distance from the most superior point on the head of the femur to the most inferior point on the distal condyles
Femur Mediolateral Diameter (FMLD)	Distance between the medial and lateral surfaces at midshaft
Humerus Epicondylar Breadth (HEB)	Distance of the most laterally protruding part of the lateral epicondyle from the corresponding projection of the medial epicondyle
Humerus Head Diameter (HHD)	Direct distance between the most superior and inferior points on the border of the articular surface
Humerus Maximum Length (HML)	Direct distance from the most superior point on the head of the humerus to the most inferior point on the trochlea
Radius Maximum Length (RML)	Distance from the most proximally positioned point on the head of the radius to the tip of the styloid process
Radius Mediolateral Diameter (RMLD)	Distance between medial and lateral surfaces at midshaft
Tibia Maximum Length (TML)	Distance from the superior articular surface of the lateral condyle to the tip of the medial malleolus
Tibia Plateau Mediolateral Breadth (TPB)	Maximum distance between the two most laterally projection points on the medial and lateral condyles of the proximal articular region

Table A.1 Measurements and definitions used in study [4]

APPENDIX B

CHECKERBOARD SAMPLE

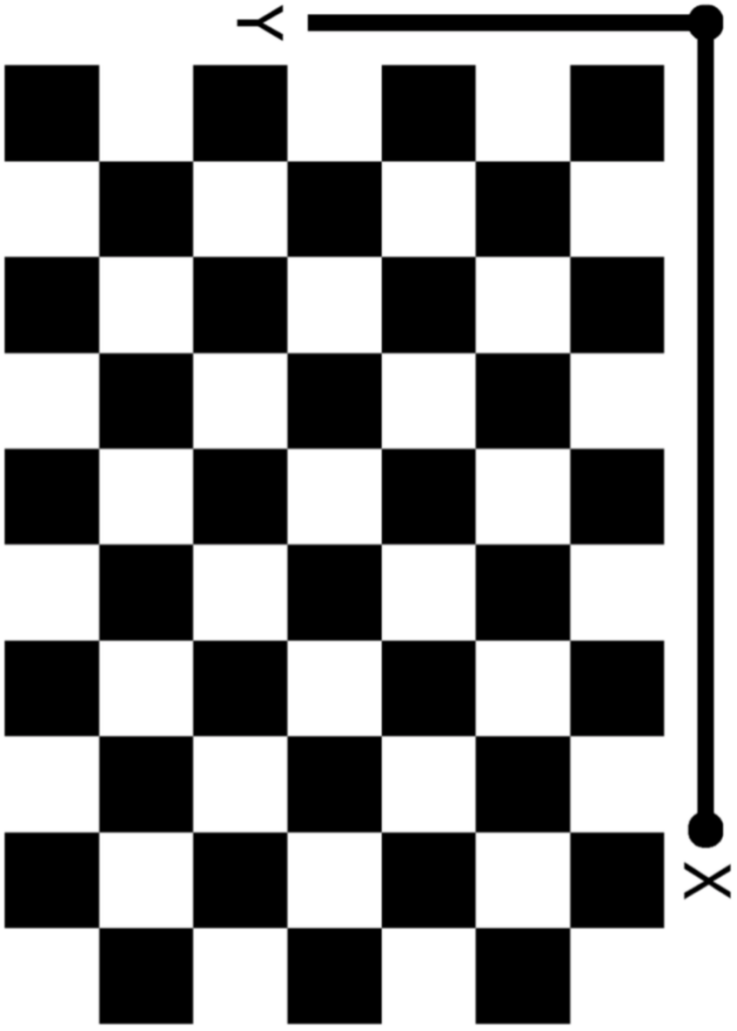


Figure B.1 Sample of the Checkerboard Pattern

APPENDIX C

LIST OF CODE FILES

The following is a description of the code used in this project. It is saved in the zip file labeled source code.

1. Filename: batchProcess.m
   Description: It is the main function to start the code. It processes all the images inside ./data folder, identifies each type of the bones from the name of the images.
2. Filename: boneMeasurement.m
   Description: Calibrate the camera or processes a single bone with a known bone type. The final measurements are saved to ./output folder.
3. Filename: adjustIm.m
   Description: Remove background noise from image by set the dark pixel values to totally dark pixels.
4. Filename: getBoneBoxes.m
   Description: Count the number of bones inside image and output bounding boxes for each bones.
5. Filename: measureVertexFemur.m, measureVertexHumer.m measureVertexRadius.m measureVertexTibia.m
   Description: They are the top functions to get the measurements from each type of the bone by calling getXXX.m.
6. Filename: getBoneMask.m
   Description: Return a binary image which precisely segments the bone region from the input images.
7. Filename: rotateFemur.m, rotateHumerus.m, rotateRadius.m, rotateTibia.m
   Description: Rotate the binary bone images so that same types of bones always have the same set of layouts, which makes the following steps possible. Also crop the binary image to a much smaller image for faster processing in the following steps.
8. Filename: transformPoint.m

Description: Transform a point in the cropped image coordinates back to the original image coordinates.

9. Filename: getFBML.m, getFEB.m, getFHD.m, getFMLD.m, getHEB.m, getHHD.m, getRMLD.m, getTPB.m

   Description: Find POIs for FBML, FEB, FHD, FMLD, HEB, HHD, RMLD, TPB.

10. Filename: getRealDistance.m

    Description: Get the real-world distance with vertexes coordinates.