

TDK-dolgozat

Formanek Balázs István



PÁZMÁNY

Pázmány Péter Catholic University

Faculty of Information Technology and Bionics

Semiautomatic Bone Segmentation and Anatomical Parameter Extraction on Hand X-ray Images (Félautomata csontszegmentáció és anatómiai paraméterek kinyerése kézröntgenfelvételeken)

FORMANEK, Balázs István (2nd semester)

Supervisors: Tasi, Benedek József (researcher and developer);
Naszlady, Márton Bese (engineering teacher)

Contents

Abstract	1
1 Introduction	4
2 Related Works	11
3 Dataset	14
4 Application	18
4.1 Application description and resources	18
4.2 Application workflow	20
4.2.1 Hand Preprocessing	21
4.2.2 Joint and Bone Localization	24
4.2.3 Bone Segmentation	29
4.2.4 Parameter Extraction	33
5 Results	36
5.1 Joint and Bone Location	36
5.2 Bone Segmentation	37
5.3 Parameter Extraction	42
6 Summary	44
Bibliography	48

Abstract

Currently available medical prosthetic devices generally lack any kind of personalization in their functional design, impacting patient comfort and intuitiveness of use, especially so in the case of hand prostheses. To improve upon this, novel digital approaches can be leveraged to tailor these devices to each user's body image. Based on anatomical rules, a 3-dimensional hand skeleton model can be constructed - to attain the desired level of individualization, the extraction of a set of patient-specific parameters is required, which is a challenging task in practice due to the lack of our ability to perform direct measurements on the bone geometries.

With advanced programming algorithms, however, we can construct complex models to process digital images, made by various medical imaging technologies (e.g., X-Ray, Computer Tomography, Magnetic Resonance Imaging). This can be applied either to assist treatment decisions or to better understand human anatomy with non-invasive methods. In this study, I introduce the Semiautomatic X-ray Processing application for bone segmentation and anatomical parameter extraction on digital hand X-ray images. This includes automatic and manual processing steps, combining the fastness of a digital algorithm with the reliability of a human expert. This kind of parameter extraction is intended to obtain the required dimensions towards the development of anatomically correct biomimetic hand prostheses.

The main research issues are to accurately separate (segment) the bones from the surrounding soft tissue and to measure predefined metrics, such as bone lengths and widths. As joint region annotation is an especially hard task which requires anatomical knowledge, the problem of bone segmentation in hand X-ray images remains unsolved. I describe a novel algorithmic solution for bone segmentation and marker point localization, that can be fine-tuned through a user interface. The application implements anatomical considerations and highly relies on biological and physical knowledge about the human hand and X-ray imaging. Using this multistage model, I performed the examination of 21 annotated hand X-ray images in around 40 minutes each, including the

whole process, compared to 2 hours which is the time it usually takes to do the segmentation only, with the help of traditional image editors. The software implements several underlying segmentation algorithms, which can even have prediction values of 90% and recall values of 99% for an image. As a result, I have obtained the predefined metrics on the segmentation masks and reached measurement outcomes similar to state-of-the-art statistical descriptions.

Acknowledgements

SUPPORTED BY THE ÚNKP-23-2 NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY FOR CULTURE AND INNOVATION FROM THE SOURCE OF THE NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION FUND



Chapter 1

Introduction

In this study, I propose an application to examine human hand X-ray images through a semiautomatic workflow. The pipeline contains fully automatic processes supplemented with user interaction. The automatic processes rely on anatomical characteristics and X-ray imaging properties. The main steps in the application are a hand segmentation preprocessing step, then the joint localization, the bone segmentation and the extraction of the anatomically defined bone dimensions. By applying the calculated parameters in the last step to a three-dimensional generic hand model we can construct a personalised hand model with the individual parameters of a person, combining general anatomical knowledge and individual patient properties. This process can be the foundation of a personalised, biomimetic prosthesis development. Copying human anatomy leads to advanced prosthetic solutions increasing amputees' comfort and the device's ease of use.

Exploring the human body through medical imaging techniques has rapidly become one of the standards in diagnostics since the discovery of X-rays in 1895 [1]. The main technologies: digital X-ray, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are in day-to-day use in clinics to help medical decision-making through non-invasive (or minimally invasive) procedures. While minimal damage is one of the common aspects, these procedures have many differences. A digital X-ray is a two-dimensional image of a part of the body (a two-dimensional projection of the three-dimensional body), usually used to detect skeletal anomalies (e.g. fractures). CT and MRI are three-dimensional "images", for further detailing the underlying anatomy of the body, these are usually used for complex, structural (e.g. organ) examinations. While CT and MRI detail the body better, X-ray is the most widely used technique because of its low cost and low radiation effects.

Without overlooking the help of these aids, it is clear that the visual examination of

the resulting images falls into human limitations and subjective decisions, that may vary based on the examiner's expertise, skills, attention and the image quality itself. Medical Image Processing developments aim to help medical staff by mitigating these limitations through digital image processing methods, with mathematically defined functions or statistical decision-making to help medical staff. Diagnosis and treatment planning should always be done by human experts but with advanced suggestions faulty diagnoses might be avoided.

Medical Image Processing has two main objectives. First, it helps medical treatment decisions and second, it provides a way of understanding human anatomy in detail. This field includes all the main image processing problems, such as classification, detection and segmentation. For example, illness classifications fall into the first category, fracture, or local discrepancy detection corresponds to the second, and different tissue or organ segmentations are part of the third branch. Although the three aims are not always separated. There are studies, in which they do organ segmentation with the premise of illness detection or tumour localisation within segmented regions of organs.

However, organ and tissue segmentation is not a task that an expert would do manually - during a visual examination - with this, we can highlight different regions that can be beneficial to the overall examination process. In my case, I can make geometrical measurements on the hand X-ray bone segmentations, to extract the bone heights and widths. So a proper segmentation method is a crucial point in my research. While there are some attempts to solve this problem, manual segmentation still suppresses their results. Image segmentation in X-ray images proposes many challenges, requiring the operator to possess considerable anatomical knowledge, in addition to the time investment that accurate results needs. My goal was to create an application to examine X-ray images without imposing requirements for high medical or user expertise and without the time needed for manual segmentation and examination processes.

While human anatomy has been described in great detail in countless books, lectures and studies, the comprehensive structure including every mechanism of the body is yet to be discovered in detail - just consider internal organ complexities such as the brain. With the spread of personalised medicine, it is not only essential to understand the general physiological mechanisms but also to be capable of deriving individual properties. Image processing techniques can be useful for both general and individual characteristic extraction. It provides a medium for detailing already-known biological features and assigning global anatomical characteristics to individuals to find personal peculiarities.

In this study, I focus on the skeletal structure of the human hand (figure 1.1.) with special attention to the finger bones and bone endings to extract patient-specific properties. The human hand consists of 27 bones, classified into carpals (8), metacarpals (5) and phalanges (14). The carpals compose the wrist region, to connect the radius and the ulna in the arm to the metacarpals. All five fingers have one metacarpal and two (thumb) or three (long fingers) phalangeal bones. The finger bones are named with an abbreviation of their type and are assigned a Roman number from I to V based on the corresponding finger, starting from the thumb to the little finger. For example, M-II is the short name of the metacarpal bone in the second finger (index). The phalanges are further classified as proximal, middle or distal according to their distance from the metacarpals. For example, MP-III stands for the middle phalanx in the third (middle), DP-V for the distal phalanx in the fifth and PP-II for the proximal phalanx in the second finger. The thumb has no middle phalanx, only proximal and distal. The joints between the bones are also named and have abbreviated names. Carpometacarpal (CMC): connecting carpal and metacarpal bones, metacarpophalangeal (MCP): between metacarpals and phalanges and interphalangeal (IP) joints between two phalanges, further specified as proximal (PIP) and distal interphalangeal (DIP) in the long fingers (figure. [2])

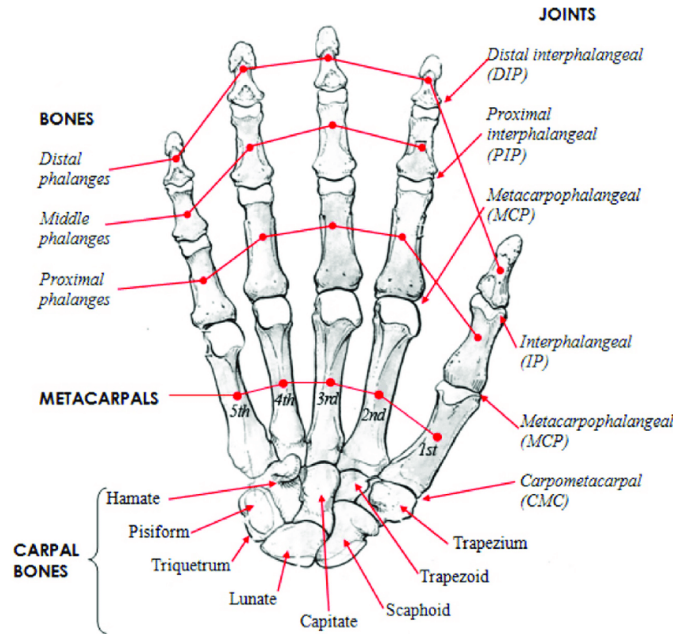


Figure 1.1: Human hand skeletal structure, depicting finger bones, joints, metacarpals and carpals [3].

The joints provide motion centres between the rigid bones, with two complementary, connecting articular surfaces. Every joint has an active and a passive range of motions,

meaning their voluntary (by tendon and muscle activations) and involuntary (externally forced) moving angles. These ranges define degrees of freedom. Finger bending and stretching are called flexion and extension in the sagittal plane, while adduction and abduction are side-by-side motions in the frontal plane. The overall structure and the possible range of motions serve the main role of the hand, the prehension. While various animal species perform grabbing, this highly developed form is only present in primate anatomy, thanks to the opposition, the thumb's special movement, to contact the tip of the thumb with the other fingers. This function is responsible for most of the operating hand movements, hence many structural specificities point to it. For example, the carpal bones and the palm are in a natural gutter formation; the metacarpals are not parallel to each other from an upper view; the second and the third CMC joints are rigid but the fourth and fifth have small flexion and extension motions; both metacarpal and phalangeal bones have a small, but not negligible sagittal arc in their shape. [2]

Regarding the possible range of motions of the individual joints, I detail here the MCP and the IP properties (note that I already described the CMC in the previous paragraph). The MCP connects the metacarpal head, which has a biconvex articular surface and the base of the proximal phalanx, ending in a complementary, biconcave articular surface which is smaller in surface area than the head of the metacarpal. Having these in common, the particular geometry of the metacarpal surfaces varies defining different ranges of motions for the individual fingers. All of them have two degrees of freedom. Flexion has a range of about 90° , with a progressive increase from the finer I to V (to enhance opposition). Active extension can reach $30-40^\circ$, with a passive extension of even 90° having very lax ligaments. Adduction and abduction are also executed in MCP joints, with the largest range in the index and little fingers. Note that although all the MCPs have two degrees of freedom it does not mean that they can be actuated independently, flexion and extension of one finger may influence the other's positions even if it is involuntary. The IP connects a pulley-shaped phalanx head with a phalanx base of two shallow facets separated by a median ridge geometry. It only has one degree of freedom, with a range of flexion around 90° (which can even be bigger in PIPs). Although all IP joints have only one degree of freedom, these are not parallel, which introduces finger obliquity towards the thumb. This is a result of the asymmetrical structure of the surfaces. MCP and IP joints have no active radial motion, although they can be forced to a small degree, which also helps prehension. Forced motions do not mean hurtful positions - our fingers can be influenced by external objects during grasping that alter

joint positions and limits when compared to purely active movements. [2]

Why do we analyse the joints to such detail? As implied before the geometrical structure of the bone surfaces (alongside tendinous and muscular activations) directly defines the executable motion directions and ranges. This means that by copying the complex bone geometries, one can construct a three-dimensional skeletal structure of the hand, that defines every possible movement by simple being based on its anatomical geometry. Leveraging this, anatomically correct, biomimetic hand prostheses can be built that mimics biology towards broader, more intuitive applicability. [4]. Furthermore, by modelling the individual parameters of an amputee’s healthy hand, we can provide a prosthetic hand fully personalized to their biology, promising better body-image integration. Prosthesis integration challenges raise important and interesting questions, connecting prosthetics and psychology. From a developmental point of view, in biomimetic prosthesis development, they do not have to define motion trajectories, with complex actuation systems and feedback loops of performed motions. Actuation should also be implemented by mimicking the anatomical tendonous and muscular activities, though in this work I do not go into detail about the prosthesis development, merely provide a data-based foundation for generating personalized prosthetic models.

In this study, I introduce an application to extract a substantial set of bone parameters from digital hand X-ray images that can be applied to a generic three-dimensional hand model for personalisation. These parameters are the bone lengths, and the head and base widths in the sagittal plane. Assuming accurate segmentations, the parameters can be defined as the following. We are looking at the bone in a vertical position the head is at the top and the base at the bottom. The head and base widths could roughly be estimated as the length of the largest radial-ulnar section in the upper and lower part of the segmentation (vertical lines), but we have to consider that the PA hand images do not grant perfectly aligned bones, hence the previously-mentioned sagittal arc should not be neglected. To estimate this arc, I propose a way to fit a second-ordered polynomial curve using a geometrical middle line between the longitudinal edges of the bone. This polynomial curve crosses the bone border at the two articulate surfaces at the surface middle point estimates. At the half point between these points on the curve, another point can be defined for the bone middle point. Note that these three points define the same second-ordered polynomial arc. The upper and lower border points should then be set to the surface geometry centres, without changing the estimated polynomial shape. Moving one of the points should not change the other, and the middle point is updated to

keep the polynom's shape. The bone length is the Euclidean distance between the surface middle points, and the widths are equal to the lengths of the longest sections inside the segmentation in the head and base regions tangent to the axle at the corresponding surface middle centres (1.2 describing figure). For these measurements in one bone, we need its proper segmentation.

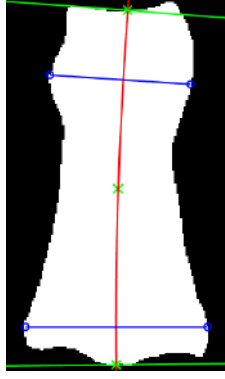


Figure 1.2: A bone segmentation showing the geometrical definition of the length and width parameters. A red plot for the middle axle, estimating the bone arc. The tangent lines, the surface and the bone middle points are shown in green colour and the head and base width sections with blue lines.

Processing all bones manually is tedious, it requires attention and medical expertise. Only the metacarpal and phalangeal bone segmentation takes two hours with basic annotation tools, without annotating the marker points. Developments in the medical imaging field help us create automatic and fast segmentation systems, but their accuracy is questionable. These models usually neglect the proper segmentation of the joint spaces, which is especially highlighted in our goal, to describe articular surfaces. A semi-automatic approach can be a good intermediate solution, to propose a fast, easy-to-use application while maintaining reliability.

I introduce the Semiautomatic X-ray Processing software, which includes automatic steps for bone and hand segmentation, joint detection, and marker point localization, closing with parameter extraction. In the application, a user can adjust the automatic predictions in several steps to maintain reliability. Although there are some attempts to solve joint localization and bone segmentation, manual examination still suppresses these solutions. There are some particularly complex problems on which automatic models tend to fail, such as finding CMC locations or segment bones following the anatomically described, highly visible surfaces. In the vicinity of the carpal bones, the exact CMC locations are hard to detect, because of the surrounding similar regions. Also, incorrect

automatic CMC localization can come from the fault assumption that all the fingers are in a straight position. By creating a two-dimensional projection from a three-dimensional object, bone shapes are not straightforward to segment. To follow the anatomical considerations described previously, it is not sufficient to separate bone-like regions from soft tissue-like regions. As of now automatic processes fail to adequately solve these problems, and my study does not aim to be the best possible solution for neither joint localization nor bone segmentation. My goal is to introduce a flexible workflow that generates estimations for the described problems, which can be improved by proper user interactions facilitated by certain function implementations in the application.

I evaluate both the automatic and the corrected segmentations created in the application compared to a ground truth dataset of 21 manually annotated images with generally used pixel-based performance metrics (derived from the confusion matrix). Finally, I compare the extracted parameters to existing three-dimensional hand model parameters, aggregated from CT images [4], with statistical descriptors. As the ultimate objective, my research work can be used in a biomimetic prosthesis development workflows to generate personalised anatomical hand models.

Chapter 2

Related Works

X-ray medical imaging is based on the primary property of the X-ray beam: it penetrates through solid substances with different characteristics, based on the material's internal structure, absorption and scattering properties. X-ray images are taken by generating a radiational beam field from a source, directing it towards the body, positioned in front of a detector. The detector platform absorbs the colliding rays that have already interacted with the body, creating a latent image of its internal structure. Hand bones absorb or scatter a larger portion of the colliding rays than the surrounding soft tissues, creating lower X-ray concentration under them, meanwhile, the soft tissues reflect less radiation, maintaining high X-ray concentration. While theoretically, the high density of X-ray beams should correspond to brighter image values, for easier visual examination they inverted this notion, so on digital images bone regions are shown with white intensities, and soft tissues with grey values. This imaging technology provides an effective, low-cost and noninvasive way of skeletal examination. (Having said this, radiation effects should always be considered.)

Although extracted images are primarily used to detect structural anomalies, dislocations, foreign object position and bone fractures [5], over the past years researchers have utilised them to predict patient age. During growth, the hand's skeletal structure and the bones' internal composition change, from which they can estimate age. Developments in bone age assessment led to much research in hand X-ray processing, including traditional [6] and machine learning-based methods [7]–[12]. Efforts have been made to automatically detect the joint locations [10], [13], measure joint gaps between the bones [14] and segment bones from the soft tissue [11], [12], [15], [16]. In 2017 the Radiological Society of North America announced the RSNA Pediatric Bone Age Challenge [17] introducing a dataset of more than 13000 hand images from patients between the ages 1 and 19. This

has become a generally used public dataset.

Andrzej Bielecki et al. developed a traditional image processing method [14] to locate joint positions, starting with an initial bone mask based on Gaussian filtering, then using thinning and searching joint locations on that structure. Gyu-Sung Ham et al. used a U-net-based method [13] with 895 images and 37 landmarks, containing the joint positions a different one, that is better/worse, because of something. After correct joint location extractions, joint space width has also been extracted by Andrzej Bielecki et al. using bone surface geometry estimations. However they did not work on bone segmentation, their way of defining the borders is the closest one to the anatomical definition I use.

Although there are some developments in bone segmentations [15], [16], these solutions neglect the joint regions and evaluate results with one segmentation mask, combining all the bone masks. In some cases not only has the method been developed without considering the joint gaps, but the ground truth data also contains the joints as bone tissue [15]. General segmentation metrics, would not show this as a problem, because the size of the joint regions is small compared to the bone size of the bones. To work on individual bone parameters, individual bones should be segmented and evaluated individually. I do not only need to take joint regions into account, but these regions are the most important ones in my scope. Without individual bone segmentations, the parameters would not be extracted. Although José M. Sotoca et al. proposed a technique [18] using mathematically defined active shape models, adjusted to every bone in the original X-ray image, they did not evaluate the segmentation, as it was a step in their mass assessment process.

As it is at the heart of many image processing developments, the rapid growth in deep learning techniques in the field of medical image processing cannot be overlooked. Although without a substantially large annotated dataset direct segmentation is yet an unsolved problem. In the last few years, there have been some promising developments. Kyunghye Jung et al. published a work on segmenting the carpal, the metacarpal and the phalangeal regions from each other in the last year (in 2023). Two further models can be found online among a large collection of open-source computer vision datasets, also published in the last year [11], [12]. However, Kyunghye Jung’s work does not directly aim to solve the bone segmentation problem, these researches show that the automatic solution to the segmentation problem is currently in progress, but is yet to be solved.

My current work is highly based on one of my previous studies on the same topic [19]. In that state, I worked with only 7 images on segmenting the IP and the MCP joints in

the long fingers, having the fingertips and the CMC middle positions as inputs. In the current proposed model, there is no need for any additional input other than the X-ray itself to start the process, rather the model calculates an initial hand segmentation and marker points. I focused on bone segmentation in contrast to joint segmentation. Finally, I extracted bone parameters based on the segmentation. I highlight the developments made in the automatic steps in the Application chapter. Other than improving the algorithmic background and model structure, I implemented a graphical interface for semiautomatic analysis and evaluated the results on a larger dataset of 21 images. This provides an easy-to-use middle ground between the convenient automatic process and the more reliable, but slower manual work.

Although many different models have been proposed for joint detection and bone segmentation problems over the years, there is no unified public dataset with a proper description and annotations. Each research group has to define their own dataset, with their proposed considerations, which makes it impossible to compare different solutions. In this study, the bone mask definition relies on anatomical structure, which makes it unique among the described state-of-the-art models. I do not use a metric to calculate joint location distances from an ideal position. As joint locations are used only to extract bone regions from the original image, there is no need to define complicated measures and ground truth data. If its position is between the corresponding two bones and the bone regions of interest are extracted correctly, the joint location is correct. Furthermore, contrary to other research, I do not introduce an optimal automatic solution for the proposed problem. Rather, I have developed a novel interactive workflow to connect automatic processes and user interactions, built into an application.

Chapter 3

Dataset

Testing an image processing algorithm in action is a crucial step both during the development and in the evaluation steps, to show model applicability and reliability. Hence an adequate set of images is always required to test the model. Although visual results are informative to the user and the developer, for objective quantitative and qualitative comparisons annotated data should be used. Annotated data includes the expected output, also known as the ground truth data. Its datatype depends on the proposed problem, it can be a class name (left or right hand), a numeric value (distances), a binary mask or another image as well. In the bone segmentation problem, the expected output is the binary segmentation of the individual bones, hence the ground truth data in this research contains 19 bone masks for every image. I also work on left-right hand detection, joint localization and distance measure extraction, but for the last two, there are no corresponding ground truth data stored. Besides giving a way to compare one proposed solution to the expected output, measurements between the truth and the predictions let us compare different solutions with each other, based on their relation to the ground truth.

After considerable interest in automated bone age detection on X-ray images, the Radiological Society of North America announced the RSNA Pediatric Bone Age Challenge in 2017 [17]. To compare the participating teams' efforts they introduced a dataset of more than 13000 *.png* images of hand X-rays. Since that, it has become a generally used public dataset. It contains posterior to anterior (PA) hand X-ray images of both left and right hands, from male and female patients between the ages of 1 and 19. As it was proposed for age prediction, the provided metadata contains the patients' age in months with their sex. Although based on this dataset, there have been researches with annotated data in bone and soft tissue separation [15] and joint location detection [13],

the applied ground truth labels are either private or if public, they are not sufficiently documented, and do not follow the described anatomical considerations [10]–[12]. Therefore I hereby propose a set of 21 manually annotated images of adult patients, for model evaluation.

The main purpose of this research is to find defined anatomical features. General anatomical rules are usually described for adults, as in the case of a statistical analysis of bone length by Alexander Buryanov and Viktor Kotiuk [20]. Therefore I needed annotated adult patient X-ray images with some variability in patient and image characteristics, to imply anatomical properties. So my dataset contains 21 X-ray images, with the following considerations. They are from both male (20) and female (1) patients, ages of nineteen (12), and eighteen (9), left (20) and right (1) hands. In all of the images, three requirements should stand. First, the hand is in the middle and separated from the image frame (it does not touch or is not close to the frame). Without this requirement, the hand masking step would not work. Second, all the bone articulate surfaces are visible, bones do not intersect to a degree that the surfaces are not visible. Third, the fingers are not touching each other, every valley between any two fingers is visible. Without this, the fingers are not separatable from each other. Images on which any of the fingers are close to the image edge, or the bones or the fingers are overlapping were sorted out (figure 3.1.).



Figure 3.1: Four adult images from the RSNA dataset. The first is an acceptable one, that has been used in my data, completing all the stated requirements. In the second, the thumb touches the frame. In the third one, the thumb phalangeal bones are not separatable from each other, the IP joint surfaces are not visible. And in the fourth image the fingers II, III and IV are so close to each other that the valley regions are not detectable.

In a previous state of this study, another requirement was, that the fingers should be straight [19]. This was considered to be a legitimate requirement, because in anatomical

theory, while the fingers are voluntarily spread out the phalangeal and metacarpal bones are aligned [2]. However, in practice, this seems to be a false assumption almost every time. The patients themselves can force their hand posture or the examiner can change the finger positions, changing the posture from voluntary to involuntary. Also, it is not certain that the X-ray beams are perpendicular to the hand. All of these individually and together can lead to bent finger images, which makes joint location detection harder. Automatic models may not be robust enough to deal with this problem, but with semi-automatic interactions, users can adjust incorrectly detected joint positions, so there is no problem in processing such images in the application.

In addition to the differences in patient attributes, the image sizes and the hand position also differ between the images. This introduces data variety to test the model in different scenarios, measuring its robustness. The automatic part of the model is required to be adaptive to variations in pose and size. This for example means that when a bounding box for a bone should be used, that cannot have fixed sizes, but should be calculated from the image.

There is no other stored metadata about the image or the patient, for example, the machine that has been used to create the image, any machine settings, the image resolution, and so on. This last one is particularly important, because, without any defined resolution, we cannot convert image measurements (taken in pixels) to real-life metrics (e.g. millimetres, centimetres). As it is going to be detailed in the next chapter, the application provides the possibility for this conversion based on a resolution number, to have the measurement in millimetres. Although, in this study, the majority of the evaluation is in pixels.

After pruning the dataset with the above-mentioned requirements, I manually created binary segmentation masks for the 19 bone fingers for the 21 images using GIMP GNU Image Manipulation Program. For every image, there are 19 masks, with the same size as the image, with black (0) and white (1) pixel values (figure 3.2.). White pixels correspond to the bone and blacks to the background. Having 19 different masks helps to differentiate between bones, and in some cases at the CMC regions, metacarpal bones overlap, which makes it impossible to have only one mask containing all the segmentations.

While automatic segmentation is not a solved problem, because of its complexity, there are challenges in defining and creating manual segmentations too. Segmentations should differentiate between the bone regions and the surrounding soft tissue, considering the underlying anatomical characteristics. Although this is true, it is not easy in practice.

On one hand, head and base surfaces do not always clearly show the anatomical geometries. On the other hand, concave three-dimensional structures may not create concave projections in a two-dimensional image. Structures in the CMC region introduce further difficulties, because based on the image quality and the hand itself, bones may overlap, showing larger or smaller parts of the real bones. These make segmentation more complex (either in manual or automatic ways) and require the annotator to possess substantial anatomical knowledge.

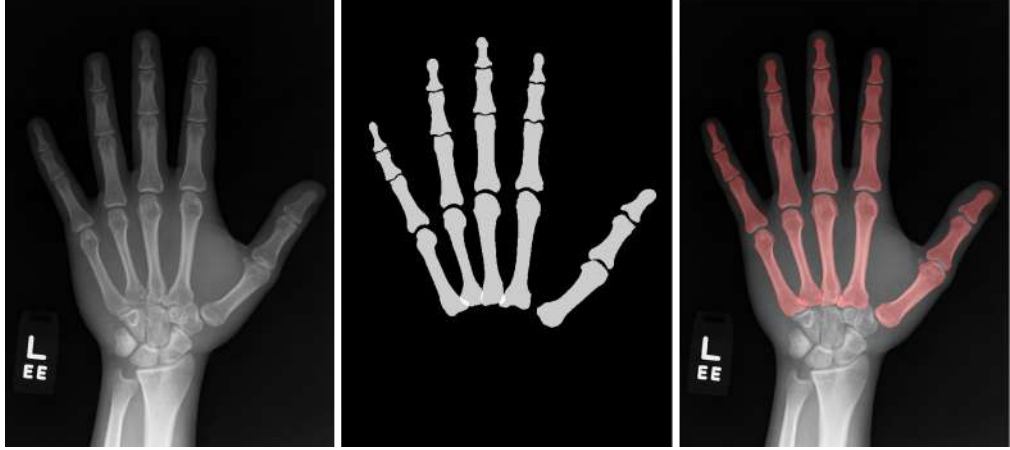


Figure 3.2: An image from the dataset with its binary bone segmentations. In the second image, the segmentations are shown with an opacity of 0.8, to show the overlaps. In the third image, the segmentations are shown with transparent red on the original X-ray image.

Using the described dataset, I evaluated the automatic and semiautomatic steps described in the next chapter, to give an objective, qualitative description of the application's power.

Chapter 4

Application

I introduce a novel semiautomatic application based on my previous work and the described studies in the second chapter. This application implements a semiautomatic hand X-ray processing workflow to find finger joint locations, segment bones and propose bone parameters, with automatic and manual steps. The automatic model has four main steps, to decrease the processed region from the whole image to the hand, then the bone regions, and finally to the bone geometry. After every main step, the user can adjust the partial results before the next step.

1. Hand Preprocessing: propose a hand mask and its boundary based on local entropy image to exclude background in further steps.
2. Joint and Bone Localization: find characteristic marker points on the hand boundary, cut fingers from the original image and propose joint locations in the fingers, based on bone length statistics to cut bone regions of interest (ROI). The user can adjust the extracted joint positions, and then extract the bone 19 ROIs.
3. Bone Segmentation: segment the bones on the ROIs based on local properties.
4. Parameter Extraction: manually correct the binary segmentations, define marker points and extract bone lengths and widths based on the bone masks.

4.1. Application description and resources

The application was developed in MATLAB R2023b App Designer but can also be used without a licence or MATLAB software. It is a stand-alone application, that can be installed from an *.exe* file, containing every code and data packaged with MATLAB's Application Compiler Toolbox. I use a Dell Latitude E5450 laptop, with Intel core i5-4210U CPU (Base Frequency: 1.70 GHz, Max Turbo Frequency: 2.39 GHz) and 12 GB

RAM, without a graphics card, using Windows 10 operating system, to run the software and create the evaluations.

The graphical interface (figure 4.1.) is divided into two parts. On the left side, is the processing image and on the right side the tools on different tabs to choose how to operate on the image. Every main step has an individual tab, where the user can change parameters, and run the automatic steps. In addition to this, there is a main menu with usual file options, to load an X-ray image and save the results as mask images or MATLAB files. There is a log line to inform the user about the ongoing operations and warnings or errors, the cursor's current x and y positions (in image coordinates) are always displayed in the bottom left corner and there is a mouse action selector, to choose between manual mouse operations.

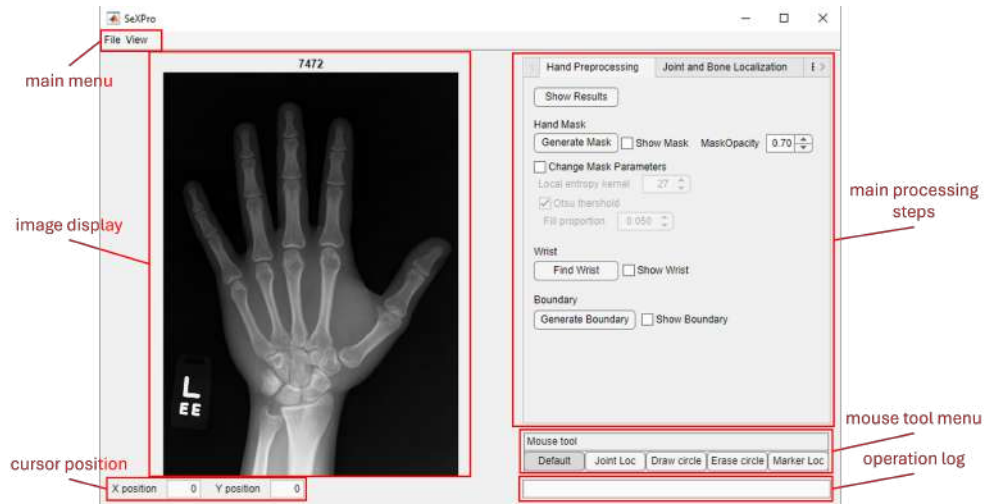


Figure 4.1: Semiautomatic X-ray Processing Application interface at the start, with the default X-ray image. The interface is divided into two main parts, the image display at the left, and the tool menu at the right, subdivided into a tab menu with the automatic processes and a mouse tool menu. The user can load and save data and change the image view in the main menu. Also, there are two additional feedbacks to the user, about the ongoing processes displayed in the operation log and the current cursor position, displayed in the bottom left.

To use the application one should install and run the Semiautomatic X-ray Processing Application from GitLab (<https://dev.itk.ppke.hu/formanek.balazs.istvan/semiautomatic-x-ray-processing-application>). It opens a valid image as a default from the dataset, but in the file menu, the user can import their image from an image file, or a *.mat* file previously saved from the software, storing a Hand object. Later one can be seen as a project file. Following the tabs on the right side of the software,

the user can go through the main processing steps, and adjust the partial results. After everything is done, they can save results in three ways: the segmentations as images, the Hand object as *.mat* files and the extracted parameters to a *.csv*.

In addition to the built-in MATLAB figure tools, the user can navigate in the displayed image, with the options listed in the View menu, or by using the keyboard shortcuts. The implemented transformations are zoom in (E), zoom out (Q), rotation left (Y) and right (X) directions, four-directional motion (up-W, left-A, down-S, right-D) and a restore (R) function to set back the original view.

For more comfortable usage, I introduced some additional keyboard shortcuts listed here:

- Ctrl+Z: undo. Before every operation, I store the previous Hand object inside the Hand class, which is used to set the object, if undo is called.
- Ctrl+Y: redo. Before every undo operation, I store the Hand object's state inside the Hand class, which is used to set the object, if redo is called.
- Shift+J: select joint location corrector mouse tool.
- Shift+E: select eraser mouse tool.
- Shift+D: select drawing mouse tool.
- Shift+M: select marker location corrector mouse tool.
- 1 or up arrow: reduce drawing circle size.
- 2 or down arrow: enlarge drawing circle.

4.2. Application workflow

After a hand X-ray import of *.jpg* or *.png* files, the semiautomatic workflow can start. The user can go through the whole process step-by-step, with manual and automatic operations. In this section, I explain the steps with the possible user interactions. The image is immediately changed to grayscale if the reading from *.pgm* or *.jpg* files results in multiple-channel images. The input is already digitalized to an image format, so there is no need for X-ray digitalization, only RGB to grayscale representation conversion, with double pixel values between 0 (black) and 1 (white). This image is then given to a Hand class object, that implements every function and stores every hand parameter throughout the process. It has fields of the extracted fingers, joint location and bones, that are members of a Bone class, which implements bone-level manipulations and characteristics.

1. Hand Preprocessing

The first step aims to segment the hand region from the background. Segmentation is done with a binary mask containing only 0 (background) and 1 (foreground) pixel values. This helps further processes, to exclude unimportant regions, like the background or the markers used by radiologists. Hand masking is based on the X-ray’s local entropy image, it is a common way to find objects in images. Entropy is a measure that shows a state of randomness, or disorder. We can calculate an image’s entropy using equation 4.1, where h is the distribution of pixel values on the frame (histogram vector) and $\odot$ denotes the elementwise multiplication of two vectors. The local entropy image is calculated by assigning the entropy of a neighbourhood around every pixel. This neighbourhood is defined as a square region with an adjustable size, set to 27 by default. With smaller kernel sizes, we get noisier images, because the method is more sensitive to small local changes. With a larger kernel, the image is smoother, because it aggregates information from larger regions (figure 4.2). To keep the pixel values between 0 and 1, I normalised the results using the equation 4.2, where I is the matrix representation of the image, $\min(I)$ and $\max(I)$ are the minimum and maximum pixel values and I_{norm} is the normalized image (also called min-max normalisation).

$$S = - \sum (h \odot \log_2(h)) \quad (4.1)$$

$$I_{norm} = \frac{I - \min(I)}{\max(I) - \min(I)} \quad (4.2)$$

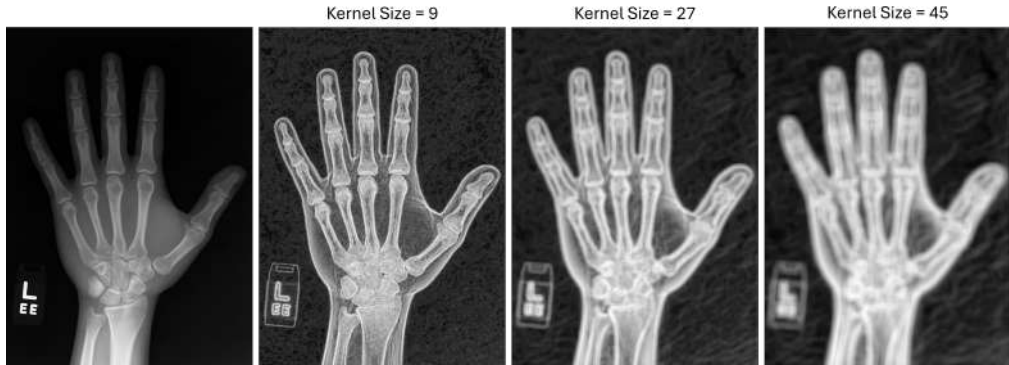


Figure 4.2: Normalised entropy images of the X-ray shown on the left, with square kernels of sizes 9, 27 and 45.

To create a binary hand mask from the entropy image, I used Otsu thresholding [21]. Although the user can define any specific threshold value to create a mask, Otsu’s method

gives a good and fast result without parameter tuning (figure 4.3). Smaller threshold values may introduce background noise into the binary mask, and higher values may not detect the finger borders, only bone regions. Otsu’s method is used in many image-processing tasks for foreground-background segmentation. It finds a value between 0 and 1 to separate the pixel value into two classes by minimizing the intra-class variance (or equivalently maximizing the inter-class variance).

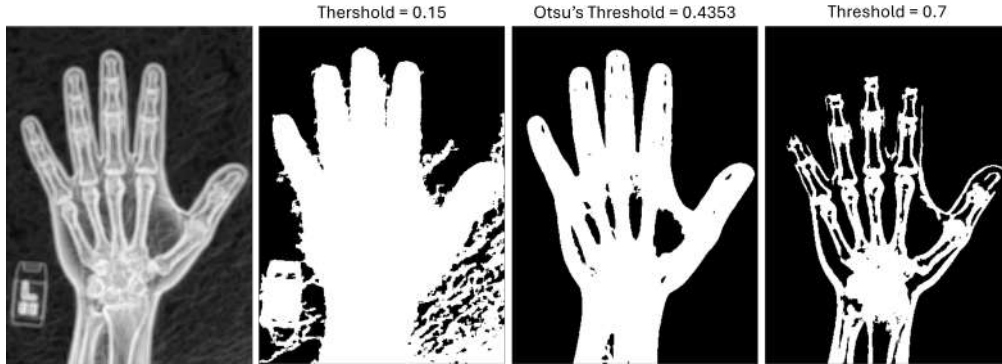


Figure 4.3: Thresholding the normalised entropy image calculated with a square kernel of size 27 (first image), with threshold values of 0.15, 0.4353 (defined by Otsu’s method) and 0.7.

Thresholding might create multiple components on the image. Assuming the hand corresponds to the largest, I neglected every other one. This assumption is based on the entropy image property, it highlights regions with large differences among the local pixels, like the hand edges, where the inside of the hand contains lighter values, than the background. Two components are separate if they do not have any adjacent (white) pixels in edges or corners. This is called 8-connectivity. They also define 4-connectivity in images, which means a component contains every (white) pixel connected to it with a pixel edge [22].

Keeping the largest component does not guarantee that it has the whole hand and only the hand in it. Firstly, although we can use the previous assumption, that the hand borders are on the mask, it does not assure that all the regions inside are on the mask, it can have holes. Secondly, the mask may contain other than the hand connected to it. This generally comes from the fact, that some images contain an image frame in them, which can result in a frame around the hand on the mask (figure 4.4.). Filling the holes would be an obvious solution for the first problem, but it would fall into the second problem, and create an unusable mask. To avoid this, the user can define a ratio number for the maximum filling area, compared to the whole image area, as default it is 5% of

the original image.

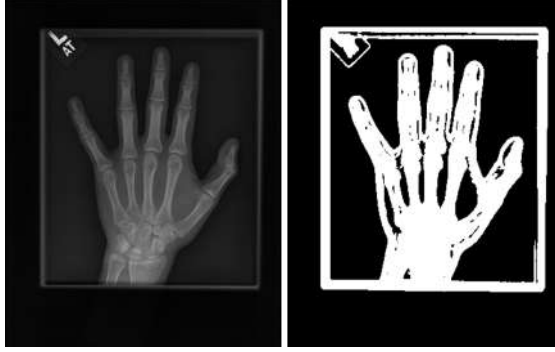


Figure 4.4: An example is that after binarizing the entropy image, it can have holes inside and a frame around it.

After the hole filling, the possible frame should be deleted, keeping only the hand. For this, I propose a wrist localization procedure, to keep only the regions above. This wrist point will be helpful in future steps, but as of now, it defines the hand's starting point. Moving downwards from the geometrical centre of the binary images (which is the average of the white pixel coordinates), in every row I define a width between the first and the last non-zero pixels until there are white pixels in the row. In the end, I set the wrist position in the row having the smallest width, in the middle of the first and the last non-zero pixels. I cut the mask with a horizontal line going through the wrist and deleted the frame. At this moment, the segmentation contains the hand and only the hand. In the last step of the Hand Preprocessing the mask's boundary is extracted with MATLAB's `bwtraceboundary` function, to a stack storing the boundary positions, starting from the closest boundary point to the wrist on the left side, going in the clockwise direction.

In the application the user can set the defined parameters: local entropy kernel size, threshold value to binarize the entropy image (exact value between 0 – 1 or Otsu's method), filling area ratio. With these parameters, the algorithm can run, and then the user can display the results on the image. The mask is an overlay, with adjustable opacity, and the wrist location and the border are two plots (figure 4.5.).

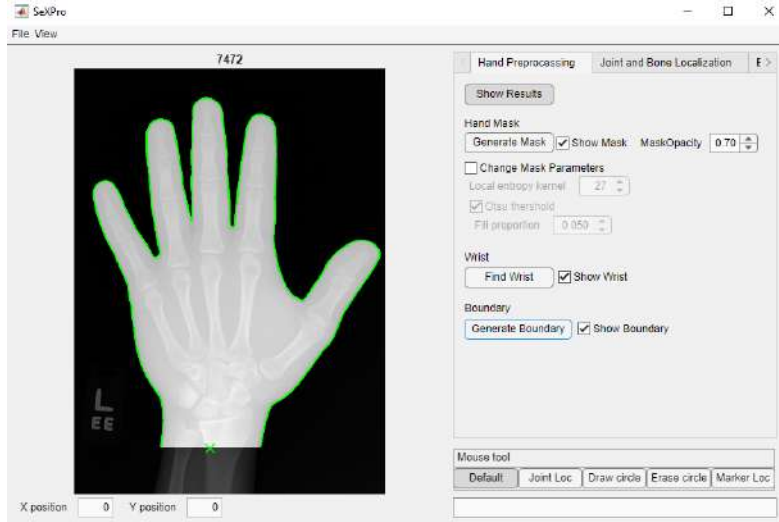


Figure 4.5: Semiautomatic X-ray Processing Application interface at the end of the Hand Preprocessing step, showing the mask on the X-ray image with 70% opacity, with the boundary and the wrist location in green.

2. Joint and Bone Localization

The ultimate goal of this step is to automatically propose joint locations, manually adjust them and extract one region of interest (ROI) for every bone. To do so, I set marker points using the border, to crop the five fingers from the X-ray followed by a joint localization technique based on Multiple Gaussian distributions, using known bone length statistics.

On the hand mask boundary, I search for notable marker points: fingertips, which are ideally the middle point of the finger ends; valleys, which are the middle points between fingers and finger foot points, which are at the middle of the finger between two valleys, connected to the fingertip these define the finger's main line. If we plot the distances of the border points from the wrist point, we get a function with 5 local maxima and 6 local minima points, 4 of them are between two maxima and 2 are at the ends of the function (second image in figure 4.6.). First let us search for the maximum positions, to have the fingertips on the boundary. Then find the 4 valleys between the fingers. Although technically the outer end of the first and the fifth fingers are not valley points, we need a similar notation to these fingers as the other three, so two additional valley points have been defined. The outer end of the leftmost finger is the closest point to its already detected valley point on the boundary section from the beginning to the first fingertip. This is going to be the first valley point. The sixth point is calculated similarly on the other side (third image in figure 4.6.). While valley points are locally the closest ones to the wrist, two of these on the same finger can have very different distances from the

tip, especially on the second finger. So I defined one additional type of marker to help with foot point detection. This I called finger bottoms. From every fingertip I checked which of the two valleys was closer, and measured this distance to the other side. Having these, the foot points are defined as the middle points between two bottom points in every finger (fourth image in figure 4.6.).

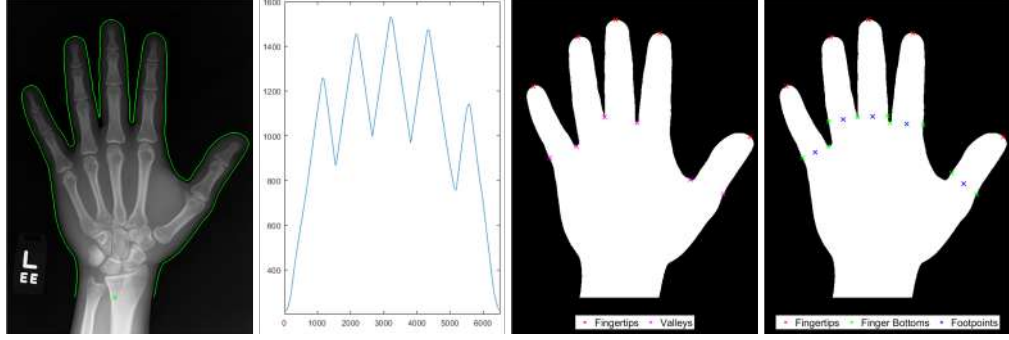


Figure 4.6: The marker localization steps. First image: the extracted boundary and the wrist position in green. Second image: the distance function of the boundary from the wrist position in pixels. Third image: the hand mask with the fingertips (red x) and the valleys (magenta x). Fourth image: the hand mask with the fingertips (red x), the finger bottom points (green x) and the foot points (blue x).

Connecting the fingertip and foot point markers, a middle line can be constructed for every finger, the finger widths can be estimated with the distances between each two bottom points and the fingertips mark the end of every finger. This is enough information to define a bounding box for every finger and cut them out of the X-ray. The width of the bounding box (the finger width) is the distance between the two bottom points. It starts at the fingertip and goes toward the foot point. The bottom end is the mask's last non-zero row when the finger is cut out from the X-ray. Based on the longer dimension of the bounding boxes we can decide if the hand in question is a left or a right hand. Among the two boxes at the side, the smaller corresponds to the thumb, from this on, we can refer to the fingers using their anatomical nomenclatures.

To find the joint locations, I use an updated version of the intensity-levelling and the Gaussian distribution-based search introduced in my previous work (figure 4.7.) [19]. I search for saturation locations on the mean intensity function, which is the row-wise average function in a finger bounding box (in horizontal alignment), excluding the background (not part of the hand mask). Saturations can be found by looking for maximum places on the absolute derivative of the mean intensity function. But instead of doing directly this maximum search, I propose a new method based on the published hand bone

statistics by Alexander Buryanov and Viktor Kotiuk [20].

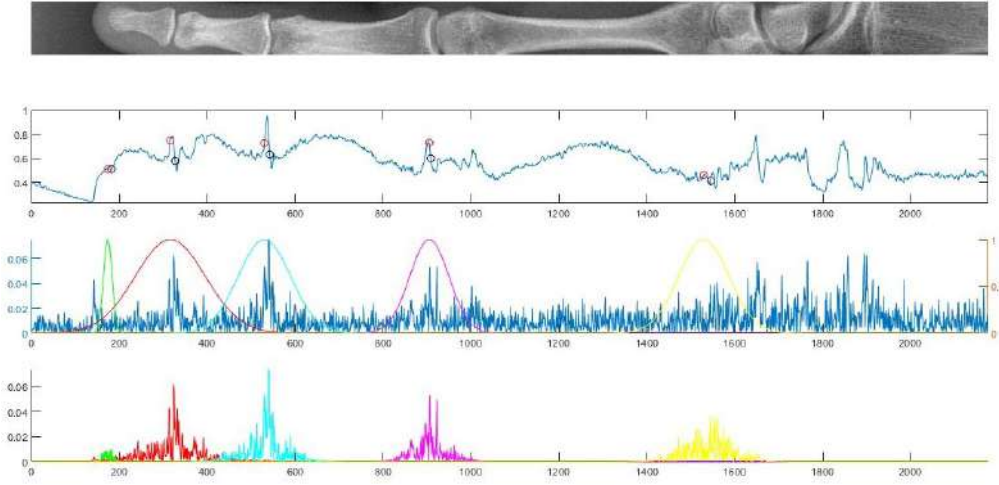


Figure 4.7: Probabilistic joint location search introduced in an earlier work. At the top the cut finger. In the second figure the mean intensity function and the found joint locations. In the third figure the absolute derivative with 4 Gaussian functions at the joint locations. In the last figure, the mean intensity absolute derivative function is multiplied by the corresponding Gauss curves from the third figure.[19]

Before the enhanced probability-based joint location search, the user can perform the same intensity-levelling method as before [19]. This introduces a vertical correction in the finger, which aims to correct the intensity decrease coming from the finger thinning towards the fingertips from the wrist. Correspondingly the soft and bone tissues are thinning, creating a gradual change in the intensities in the same tissue type. Intensity levelling estimates this change with a linear function, aggregated from fitting a linear function to every column intensity function of the finger (excluding the background with the hand mask), and corrects the finger with this linear estimation, to have similar intensities in the same kind of tissues, hence flattened mean intensity function.

The main automatic step in this section is the joint location proposal. For this, I describe a new method to find the distance between the fingertip and the MCP position, based on a probabilistic search with Multiple of Gaussian functions. I created a parametric probability function for every finger using the statistics described by Alexander Buryanov and Viktor Kotiuk [20]. It combines four (three in the thumb) Gaussian functions with the same weight, using the statistical mean and standard deviations from the paper, proportional to an input parameter of the fingertip MCP distance. The Gaussian centres are at the end of the DP, the DIP, PIP and the MCP regions. Using a Gaussian for the CMC can introduce a false enhancement in the carpal region, influencing the whole

measurement, not to mention, that the CMC is the one most likely not to be in line with other joints, because of the hand pose. So with an exhaustive search, I calculate the best fit of the probability function to the mean intensity absolute derivative function. To decrease the computational load coming from the search, I reduced the number of possible tip-to-CMC distances. The smallest checked distance is the fourth of the whole finger length, and the longest distance is half of it. These considerations come from the bone length proportions to each other and the phalangeal region length in proportion to the metacarpal. I check the distances between these two, in every five pixel steps. The best fit corresponds to the maximum value by summarising the elementwise multiplication of the Gaussian and the absolute derivative functions. The Gaussian enhance local regions in the absolute derivative functions, and when it best fits the function, it has the maximum product. Using the optimized tip-to-MCP distance the other joint locations can be set. However, the bone lengths give a good estimation of the locations, it is not certain that only the mean values apply to every patient, so based on the standard deviation I made another search around the proposed locations.

The next step was also introduced in my previous work with some modifications. To find the best joint location between two bones I use a border calculation technique. This border aims to separate the two bones in the joint region. Starting from a point on a joint image cut from the finger, a border is set by going in the left and the right directions following the minimum intensity pixels. For example, in the beginning, the algorithm adds the pixel with the smallest pixel value from the 3 pixels at the right of the starting pixel (connected in the edge or corner). It continues to propagate to the right, until the end of the image, then repeat the process to the left from the initial point. If starting from an outer point this process does not go into bone regions, because the soft tissue always has smaller intensities. To compare borders with each other I calculate the sum of the pixel intensities on the finger (or on the intensity corrected finger) under the border. The border that has the smallest sum will be the proposed one. For one joint location correction, I calculate the borders in the middle column in the finger around the previously extracted mean value with a neighbourhood of the size of the standard deviation.

After the joint locations have been calculated, in the last steps of this section the locations should be transformed back from the calculations made in the finger images, to the X-ray image, these locations then can be adjusted by the user, and the bone regions can be extracted. The bone ROIs have the same width as the finger that they are in.

The adjacent two joint locations define the middle line in the box, but the box height is slightly larger than the distance between these two points. I recalculate the separators at the joint locations and cut the finger so that it contains the whole separators. This assures that even in a concave bone ending the boundary box contains the whole bone.

In the Joint and Bone Localization step the user can go through the process step-by-step (find markers, crop fingers, correct them, find joints and extract bones). Meanwhile, they can visualize the extracted marker points, show the finger bounding boxes on the image display, or have a new figure showing the five fingers or the corrected fingers. After the joint detection, the user should check the positions, and with the "Joint Loc" mouse tool, by a drag-and-drop behaviour change the locations if needed. Based on the joint locations they can extract the bone ROIs and similar to the fingers, show them in a separate window (figure 4.8.). If the number of joint locations in the fingers does not match with reality (5 in the long fingers to define 4 bones, and 4 in the thumb to define 3 bones), the left-right detection may have failed. In that case, the user can overwrite this with a button click.

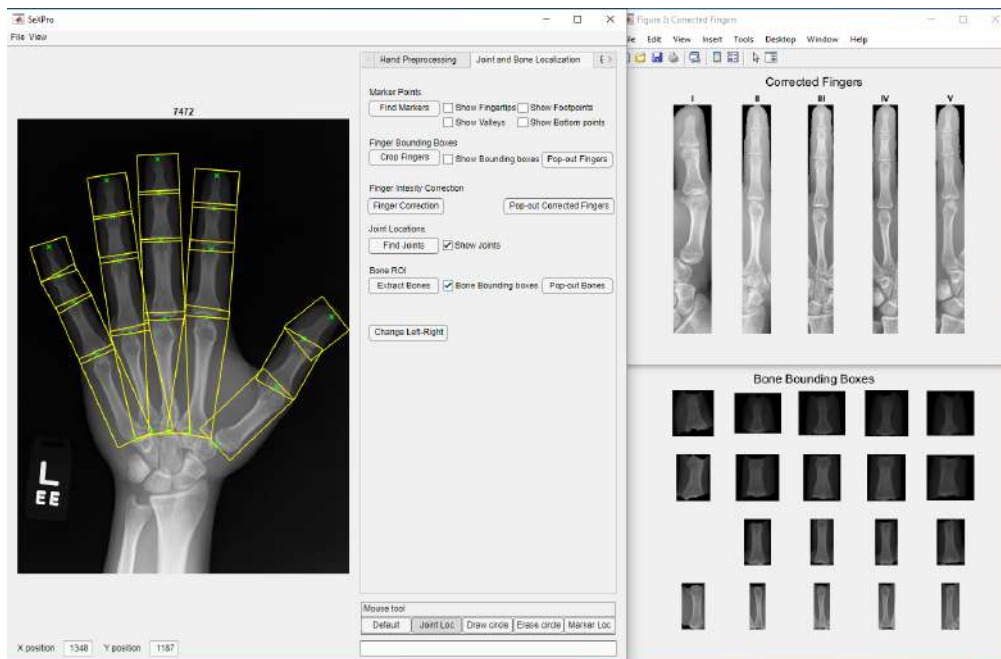


Figure 4.8: The results of the Joint and Bone Localization step. On the left side, the application interface is visible with the "Joint and Bone Localization" tab open, showing the joint locations and the bounding boxes on the image display. On the right side, the corrected fingers and the bone ROIs can be seen in separate windows.

3. Bone Segmentation

Bone segmentation is the last main step before parameter extraction. While the problem of properly segmenting the bones according to the previously described anatomical considerations remains unsolved, the application provides some possibilities for a baseline to be adjusted by the user. After the bone extraction, the user can try different segmentation techniques and manually correct the chosen one. This reduces the time spent on segmentation compared to traditional image manipulator applications.

After ensuring all the bones are fully in the bounding boxes at the end of the last step, Bone Segmentation starts. The following steps are bone ROI-level operations, which means they run independently for every bone. The previously described Hand class object contains a field with 5 Bone class objects. They store bone parameters, like the bounding boxes, the segmentations and the later extracted parameters, as well as the bone-level function implementations, such as segmentation extraction, manipulation or parameter measurements.

Generally, segmentation procedures come in two kinds, either based on the shape (like edge, structure or geometrical contour) or pixel intensity (like colour, grey level or illumination). In the middle of this step, there stands an intensity-based thresholding, but operations before and after that may vary and generate different results. In this section, I list all the parameters that can be chosen to create the initial bone segmentations for manual correction.

As described earlier, the bone ROI contains the border generated from the two joint locations at the two ends. Based on these borders we discriminate between the narrowed ROI (called ROI) and the complete ROI (called CROI). The CROI includes every pixel in the bounding box cut from the original x-ray image, without any other change. While on the narrowed ROI, the pixels above the upper border or below the lower border are set to 0 (figure 4.9.). This can help if the user wants to exclude the adjacent bone regions.

The ROI is only a part rotated and cut out of the image, so it can be advisory to normalise (figure 4.9.) it before thresholding. Although a linear normalisation may not be useful before many threshold methods, because it would not change the result (for example in Otsu's method), some used processes (detailed later) combine the image with other extracted images. Normalisation provides a way to do this combination with equal weights. The method is the same min-max normalisation introduced earlier (equation 4.2.), excluding the zero values. The last possible modification before the thresholding is an intensity-levelling, to mitigate the intensity changes in the same tissues coming from

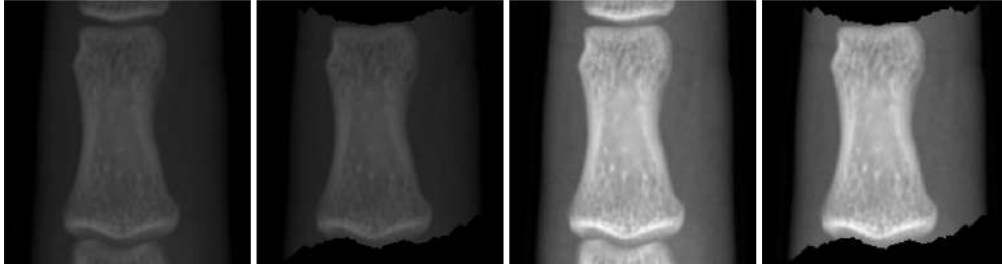


Figure 4.9: A phalangeal bone ROI cut from an X-ray image. The complete ROI (CROI), the narrowed version by cutting it with the borders and the normalised versions of these two.

the finger thinning. This was described in the previous chapter, only here we apply it to a bone, not the finger.

For the centring threshold step, five possibilities are implemented in the application. First: Otsu’s method. While it might not be an advanced option, it often gives a good result in segmentation problems, gives a fast, useful baseline. Second: bimodal thresholding. This is based on the assumption that the histogram of the ROI is composed of two main parts and can be separated by the value at the histogram minimum between the peaks [23]. In the images, there are only bone and soft tissue regions, because the background has already been eliminated in a previous step, so this assumption can stand. Do not forget, that because of x-ray and bone composition properties not every pixel inside the image has a larger value than the pixels outside, but generally, the assumption stands. To find this minimum position on the histogram, I use a five-step process (figure 4.10.). Calculate the histogram with 50 bins (in a normalised image this means the bins are: $[0, 0.02)$, $[0.02, 0.04)$, ..., $[0.98, 1]$). Cut the beginning: to ensure that the histogram has two modes, set every bin value to 0 which is bigger than the following one (including a high value at the first bin, because of the background zeros). Use sliding window-based average filtering (with size 5) to smooth the histogram function, before local extrema search. Find the two highest local maxima, and finally the minimum bin between them. (Every bin corresponds to an interval, so when one value of the bin is referred to, it is the beginning of that interval.)

The remaining three threshold possibilities are combining the bimodal method with a local variation enhancement step. Local differences can be enhanced in many different ways. We have already seen a possible way with the entropy image. Similarly, we can calculate the local range image, in which we assign the range in the neighbourhood for every pixel, and the local variance image, in which the local variance is determined. To

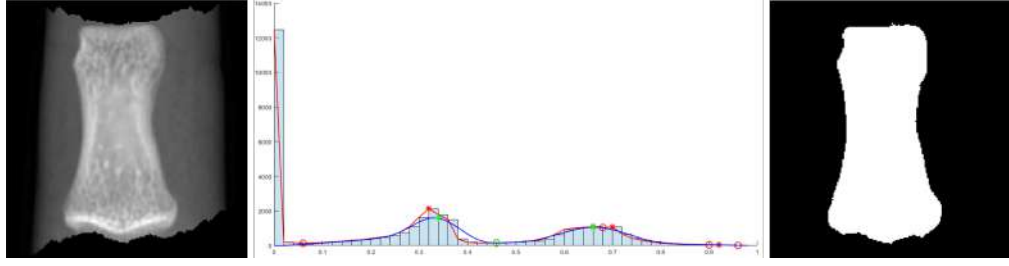


Figure 4.10: Bimodal segmentation of a normalised phalangeal bone ROI. On the left side the normalised ROI. In the middle, there is a bar plot of the histogram of the ROI with a light blue colour. On top of that the original histogram as a function and the function after clearing the beginning and average smoothing are plotted with red and blue lines. The local minimum places are marked with circles and the maximum places with asterisks, with the same colours as the function it has been calculated on. The final two maximums and the minimum between them are displayed in green. On the left side, the bone mask is shown by thresholding the image with the final minimum value equal to 0.46 (marked with a green circle).

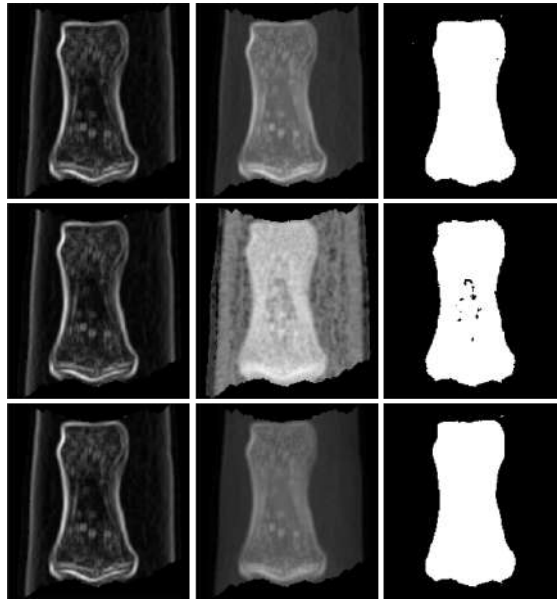


Figure 4.11: Three different ways to enhance local differences. In the first column the local variance, entropy and range images. In the second column, these three combined with the original phalangeal bone ROI, to enhance local changes. In the third column the bimodal segmentations with threshold values 0.2907, 0.582 and 0.2541.

enhance local changes on the image, but also keep information of the original image, we can add them together, after normalising each, to ensure the same weights for the two parts. The bimodal threshold is then calculated on that combined image (figure 4.11).

While the local variance, entropy and range images may seem very similar to each other, they highlight different regions with various degree, that can be seen on the combined images.

As previously in the hand segmentation phase, after a pixel intensity-based method, it is not sure that the proposed mask has one component without holes inside. So two additional postprocessing steps are provided, to keep the largest component and to fill the holes. By all these, we have the automatic segmentation proposals, to be adjusted manually.

Regarding the user activities in the step, first, they have to ensure that all the bone bounding boxes contain the whole corresponding bones (at the end of the previous step). Then they can set the preferred segmentation process: to use the CROI or just the narrowed ROI; whether to normalise the image or not; correct the intensity change coming from finger thinning; set the main thresholding method; define the neighbourhood size, which is set to 5 as a default (if the method uses local calculations); either keep everything or just the largest component and whether to fill the holes on the mask. After generating all the segmentations, they can display them on the X-ray image, with adjustable opacity, and show them in a separate window (figure 4.12.).

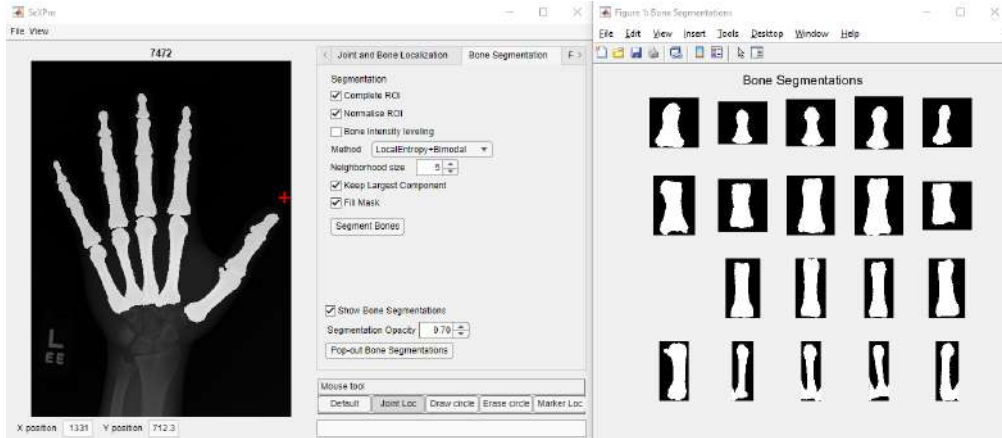


Figure 4.12: Semiautomatic X-ray Processing Application interface after the Bone Segmentation step. On the left side, the application interface is visible with the "Bone Segmentation" tab open, showing the extracted segmentations on the image display using the normalised CROI, without intensity levelling, the local entropy image with a rectangle neighbourhood of size 5, and keeping the largest component without holes. On the right side, the bone segmentations in the corresponding ROIs can be seen in separate windows.

4. Parameter Extraction

Parameter extraction starts with manual segmentation corrections. The majority of the user interactions are implemented in this step, the user can adjust the bone masks and the three marker points extracted for the bone axle. So although logically the manual correction could have been part of the previous step, it is more convenient to have every adjusting procedure available at once.

In a tree selection tool, the user can choose from the bone masks to show them on the display image with adjustable opacity, this way showing the X-ray and the segmentations on top of each other. Multiple masks can be visualized, but only the selected one (highlighted in the tree list) is correctable, this one is also highlighted on the image with a larger intensity value. There are two ways implemented to correct the selected mask, which can be activated in the mouse tool menu. One adds a circular disk to the mask, the other erases. By choosing either of them, the cursor changed to a red circle with a default radius of 20 pixels (this is also changeable via the interface). Setting the active bone and the circle radius, by clicking and hovering (while the left mouse button is clicked) on the image, the user can add or remove pixels from the mask and the corrections are immediately shown on the display image.

After the manual correction, there is a "Segmentation Done" button on the interface to end the drawing procedure and generate the bone parameters. All the parameters can be visualized on all the selected masks. These parameters are the mask border, the surface middle points, the bone middle, the polynomial axle curve, the tangents at the surface middle points and the width lines parallel to the tangents. The axle and the tangents are shown in only the corresponding bounding boxes, to simplify the figure.

First, the mask boundary is defined similarly to the hand boundary earlier, storing a vector of coordinates at the border of the mask in the clockwise direction. Then I select the left side pixels in the mask which all have black neighbours at the left. From these, the longest connected line composes the left side (using 8-connectivity). In the same way, the right side is defined. For every row on the image that has both left and right side pixels, I set a middle line pixel in the centre of the two sides. This line shows the bone's arc. To have a mathematical curve describing this arc, I fit a second-ordered polynomial to the middle line pixel coordinates. This gives the axle parameter. The surface middle points are the two intersections of the axle and the border, and the bone middle is the middle point between them in the polynomial. Finally, the longest internal sections that are parallel to the tangent lines are set at the top half for the bone head width and at

the lower half for the base width (figure 4.13.).

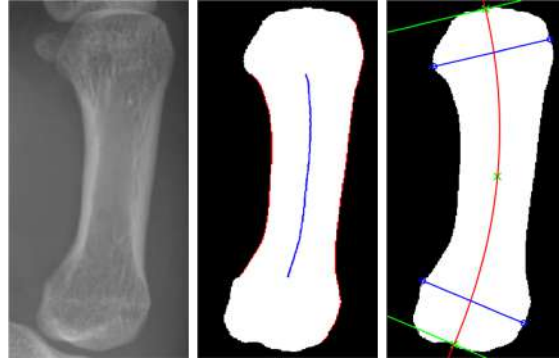


Figure 4.13: Bone parameter extraction from a metacarpal bone ROI, shown at the left. In the middle, the mask with red pixels showing the left and right sides of the mask, and blue pixels corresponding to the middle line. At the right the final parameters are visible: the second-order polynomial (coloured red), the surface and the bone middles (with green x-s), the tangent lines to the axle, crossing the surface middle points (green lines) and the sections corresponding to the head and base widths (with blue lines and circles at the section ends).

After the parameter extractions the user can change the surface end middle points similarly to the joint location corrections, there is a separate mouse tool to do so. Only one point can be changed at once, and the new position should be on the surface, which is ensured by the underlying process. The user can click on one of the two surface middle points of the active bone mask, then by hovering the mouse (while keeping the left mouse button clicked) the point position changes to the closest point to the cursor position on the border. This may not change the axle polynomial shape, and the other surface middle point, only the axle and the bone middle positions.

At the end of this step, every bone should be corrected and the bone surface middle points should be set. The user can check every parameter displayed on the image. (figure 4.14.).

After every adjustment is completed, in the last tab the bone parameters can be displayed in a table, given a resolution number in pixels per millimetre, depending on the processing image. This number is set to 1 because until this point I have only used images, without known resolution numbers, so the results are interpreted in pixels (figure 4.15.). This table can be saved to a *.csv* file for further processes.

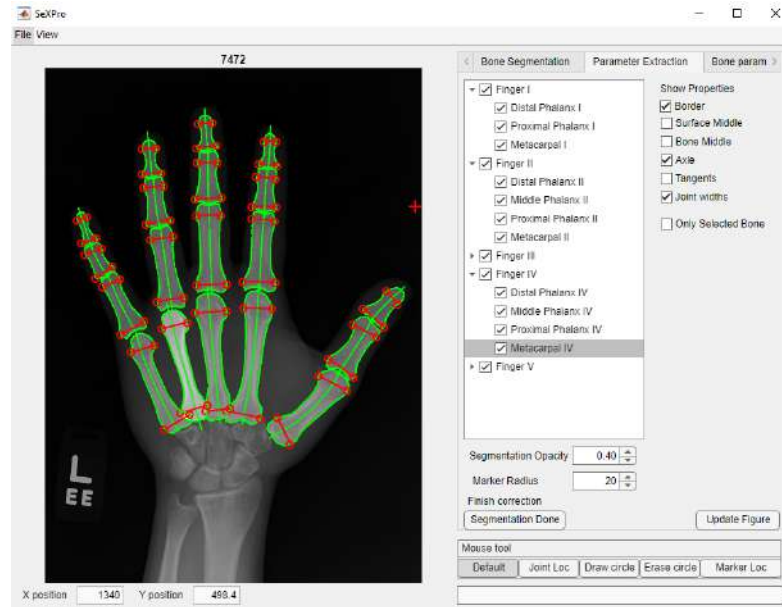


Figure 4.14: Semiautomatic X-ray Processing Application interface after the Parameter Extraction step. Showing every bone segmentation, with activating the M-IV bone. On the X-ray, the bone borders and the axles are shown in green and the head and base widths are red. On the left side, the Parameter Extraction tab is selected showing the bone-selecting tree and the display possibilities.

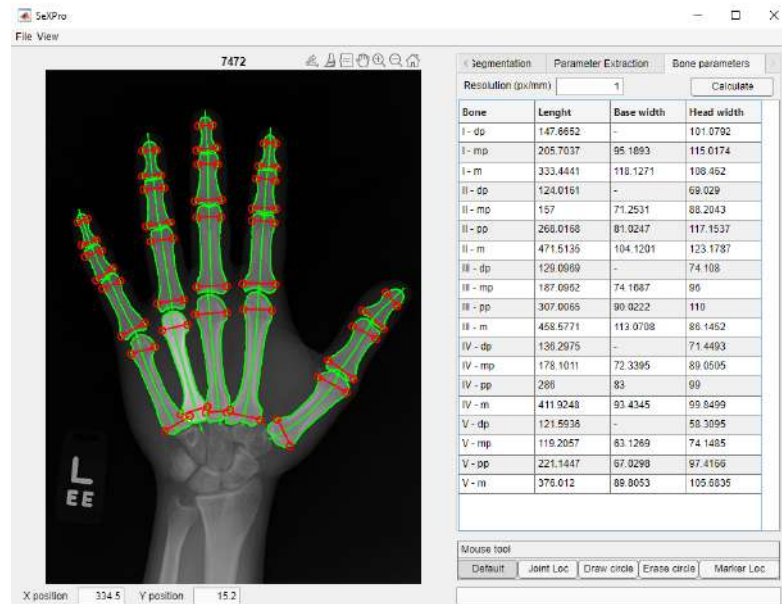


Figure 4.15: Semiautomatic X-ray Processing Application interface showing the last tab on the right, with the extracted bone parameters, and the results on the left. Displaying every bone segmentation, with activating the M-IV bone. On the X-ray, the bone borders and the axles are shown in green and the head and base widths are red.

Chapter 5

Results

To test the application and the underlying process I made experiments using 21 augmented adult X-ray images. I show the results step-by-step, as in the previous section. First focusing on bone localization, then the automatic segmentation and finally the extracted parameters. Before every evaluation step, I had to be sure that the previous processing step gave good results, or manually adjust them. This means, that before starting the automatic segmentation evaluation, I corrected the joint locations, to extract bone bounding boxes, that include the whole bones. Also, before the experiments on the parameter extraction, I adjusted the bone segmentations in 7 images, to have the below statistics.

5.1. Joint and Bone Location

Compared to other researches, we do not aim to find perfect coordinate matches when proposing joint locations. The goal of the joint localization is to propose bone bounding boxes, which contain all the pixels of the corresponding bone. So this is the aspect, I examined. I defined three categories by looking at the results of the automatic process: the ones that do not need repositioning, the ones that need to be corrected, and a third category for the locations that do not need to be repositioned to keep every bone in the box, but with the correction it gives a better box. This last category is typical in the CMC regions, the resulting box may contain the whole bone, but the location is found in the carpal region, thus creating a larger ROI than needed.

With these considerations, I checked all the 21 images, which means 399 joint locations (table 5.1.). From which 89 needed correction and I corrected an additional 18 for better bounding boxes. From the additional 18, only one corresponds to an MCP, all the others

	DIP	PIP	MCP	CMC	SUM
I	4	-	13 (+1)	11 (+2)	28 (+3)
II	3	2	4	8 (+7)	17 (+7)
III	4	2	3	5 (+3)	14 (+3)
IV	2	2	1	9 (+2)	14 (+2)
V	1	2	1	12 (+3)	16 (+3)
SUM	14	8	22 (+1)	45 (+17)	

Table 5.1: Number of repositioned joint locations corresponding to the fingers and bones examining 21 X-ray images (a total of 399 joints). Showing the number of relocations because the bone would not be visible in the bounding box, with the number of additionally repositioned locations, to better show the bones in the parenthesis. And added together by fingers and locations.

are in the CMC region. In the 89 the majority of the faulty behaviour is in the thumb (28) or on the CMCs (45), both of which can be explained by the unaligned position of the finger bones. On the X-ray image, the thumb is the only finger, that is not in a perfectly aligned top-to-bottom position, in any case, this finger is shown somewhat sideways, which can result in angels in the joints that are not treated in the automatic joint localization. Similarly, while the phalangeal bones are more or less aligned, there can be an angle in the MCP joint, because of its second movement dimension, so the CMC is not aligned with the other joints. While there are misbehaviour in other fingers and joints, these two needed most of the corrections.

In addition to the bone ROI definition, in this section, the application also gives a prediction of whether the processed hand is left or right. This decision has been good in 19 of the 21 cases and had to be overwritten in for only 2 images.

5.2. Bone Segmentation

An established way to evaluate segmentation problems is by the metrics derived from a confusion matrix. For this, we need a reference segmentation, which is the ground truth, and a prediction, which is given by the model. The number of intersecting white pixels is the true positive value because these points are part of the object in the optimal and the proposed image too. Similarly, the number of pixels in the background in both images is the true negative value. The pixels that are part of the ideal object, but are not detected

add up to the false negative, and lastly, those that are detected, but are not part of the ground truth compose the false positive value. From these two typical metrics between 0 and 1 are calculated, in an ideal model, both of which are 1. Precision shows how good the model is in selecting the pixels as foreground, the number decreases with the number of false positives (equation 5.1.). Recall describes the ability to select the background, it decreases with the number of false negatives (equation 5.2.).

$$[h!]Precision = \frac{TP}{TP + FP} \quad (5.1)$$

$$[h!]Precision = \frac{TP}{TP + FN} \quad (5.2)$$

For quantitative results, I ran the segmentation process in four different setups, using the normalised complete ROI and kept the largest component without holes. I calculated Otsu's method, the bimodal segmentation, the bimodal segmentation on the combined image with the local entropy enhancement and the same supplemented with bone intensity levelling. Then I calculated the precision and recall values for every bone and averaged these by fingers (table 5.2.), bone types (table 5.3.) and images (table 5.4.). All the tables show that the best method is combining the local entropy with the ROI and calculating the bimodal segmentation on that. The intensity levelling does not help in contrast to the hole finger intensity levelling. This supports the idea that the thinning phenomenon from the carpal region to the fingertips is a finger-level problem, and does not have that large effect on individual bones.

By averaging the results for the fingers, we can see if there is any difference in segmenting the bones in them. As in the previous section, there has been a difference for the thumb, that gave poor results compared to the others. Looking at the table columnwise it cannot be stated, that there is a finger that works much better or much worse in every method. Some fingers work better in one of the methods, and others in other methods. There is no clear sign that the methods would be finger-dependent. On the other hand, if we look at the table comparing the methods, it is clear, that the third model gives the best precision values, and the fourth gives the best recalls. This means that the entropy-based method without intensity levelling is better at positive pixel proposal, but does not produce all the pixels inside the bones, while the model with intensity levelling might give more good pixels because it has a poor background selection. Based on only this the user might decide if it is easier to draw additional parts of the mask or to erase from the large masks.

	Otsu		Bimodal		+ Entropy		+ Intensity levelling	
Finger number	P	R	P	R	P	R	P	R
I	0.7850	0.9517	0.7681	0.9420	0.8606	0.9872	0.7732	0.9884
II	0.7953	0.9397	0.7822	0.9433	0.8680	0.9887	0.8317	0.9918
III	0.8095	0.9413	0.8101	0.9581	0.8584	0.9927	0.8124	0.9933
IV	0.7793	0.9415	0.7758	0.9552	0.8236	0.9945	0.8045	0.9948
V	0.7916	0.9419	0.7529	0.9510	0.8619	0.9929	0.8274	0.9952
Average	0.7925	0.9432	0.7778	0.9499	0.8545	0.9912	0.8098	0.9927

Table 5.2: Segmentation precision (P) and recall (R) results aggregated by fingers calculated on 21 X-ray images with four different methods: Otsu’s, bimodal segmentation, bimodal segmentation on the combined image with the local entropy enhancement, and the same supplemented with bone intensity levelling.

Averaging the result by bones gives a similar view, but discriminates between bone-dependent behaviours, such as the metacarpal ROI proposal in the previous section. Looking at the table row-wise we see the same behaviour as before. The third model gives the best precision and the fourth the best recall values. The differences between these models are smaller than before, except for the DP bones. The entropy-based model without intensity correction suppresses all the other models in the DP ROIs, with more than 13%, while those three have similar results. Another outcome is although the differences in the recall values are in thousandths or then thousandths, the differences in the precisions are more harsh.

The last table shows the result through images, which gives a real overview of the power of the different models. Based on the previous table one would assume that the third and the fourth models are always better choices than the first two, this can vary in images. For example, in the 5th image, the precision values in the first and the second models surpass the other two, which can be beneficial in the manual segmentation correction.

Note that while the recall values are generally higher than the precisions this does not mean that it is a better metric, or that the models with higher recalls are better than the low precision values. High recall and low(er) precision generally mean that the model produces bigger masks than the real objects. Also note that these models are just a limited showcase of the implemented possibilities with a fixed neighbourhood size for

	Otsu		Bimodal		+ Entropy		+ Intensity levelling	
	P	R	P	R	P	R	P	R
DP	0.6934	0.9738	0.6583	0.9668	0.8232	0.9865	0.6854	0.9894
MP	0.8582	0.9782	0.8871	0.9725	0.8944	0.9960	0.8699	0.9971
PP	0.8767	0.9745	0.8681	0.9748	0.9141	0.9952	0.9122	0.9958
M	0.7549	0.8517	0.7217	0.8917	0.7930	0.9888	0.7911	0.9903
Average	0.7958	0.9446	0.7838	0.9514	0.8562	0.9917	0.8147	0.9931

Table 5.3: Segmentation precision (P) and recall (R) results aggregated by bone types calculated on 21 X-ray images with four different methods: Otsu’s, bimodal segmentation, bimodal segmentation on the combined image with the local entropy enhancement and bimodal segmentation on the combined image with the local entropy enhancement with bone intensity levelling.

entropy image extraction, so for individual images, there might be better options with a little parameter tuning.

	Otsu		Bimodal		+ Entropy		+ Intensity levelling	
Image id	P	R	P	R	P	R	P	R
1	0.8436	0.9694	0.8828	0.9629	0.8893	0.9857	0.8627	0.9918
2	0.7484	0.9641	0.6939	0.9707	0.8673	0.9896	0.7633	0.9934
3	0.7297	0.9561	0.7162	0.9686	0.7571	0.9919	0.7058	0.9906
4	0.7549	0.9787	0.7666	0.9776	0.8154	0.9939	0.7843	0.9933
5	0.7229	0.9628	0.7059	0.9752	0.6679	0.9948	0.6679	0.9948
6	0.7370	0.9590	0.6859	0.9714	0.8234	0.9893	0.7149	0.9914
7	0.8770	0.9401	0.7215	0.9624	0.9049	0.9854	0.8261	0.9883
8	0.8456	0.9405	0.7079	0.9644	0.9047	0.9858	0.7790	0.9898
9	0.5892	0.9656	0.6178	0.9750	0.7928	0.9933	0.6093	0.9953
10	0.5991	0.9399	0.6752	0.9528	0.8011	0.9915	0.6672	0.9928
11	0.8963	0.9292	0.8855	0.9448	0.8897	0.9945	0.8981	0.9944
12	0.8589	0.9137	0.8571	0.9058	0.8892	0.9963	0.8941	0.9963
13	0.8607	0.9373	0.8573	0.9332	0.8967	0.9759	0.8463	0.9888
14	0.8562	0.8978	0.7577	0.9208	0.8877	0.9933	0.8911	0.9973
15	0.8234	0.9257	0.8265	0.9323	0.8654	0.9920	0.8756	0.9910
16	0.8491	0.9182	0.8376	0.9326	0.8747	0.9946	0.8824	0.9951
17	0.8301	0.9454	0.7963	0.9544	0.8900	0.9948	0.8981	0.9913
18	0.8110	0.9173	0.8194	0.9111	0.8530	0.9954	0.8256	0.9958
19	0.8710	0.9481	0.8643	0.9562	0.9023	0.9880	0.9104	0.9878
20	0.8051	0.9552	0.8357	0.9566	0.8853	0.9983	0.8906	0.9962
21	0.7339	0.9339	0.8344	0.9283	0.8801	0.9953	0.8541	0.9958
average	0.7925	0.9428	0.7783	0.9503	0.8542	0.9914	0.8118	0.9929

Table 5.4: Segemntation precision (P) and recall (R) results calculated on 21 X-ray images with four different methods: Otsu’s, bimodal segmentation, bimodal segmentation on the combined image with the local entropy enhancement and bimodal segmentation on the combined image with the local entropy enhancement with bone intensity levelling.

5.3. Parameter Extraction

Without corresponding real data, there is a limited way to compare the parameter extraction results with true dimensions, so I used a statistical description method, to show my results. After the automatic segmentation, I adjusted them on 7 images, to extract parameters in pixels. Using the method presented in the work of Alexander Buryanov and Viktor Kotiuk (table [20], I also divided the lengths (table 5.5), the head widths (table 5.6) and the base widths (table 5.7) with the DP lengths, by finger and then calculated the means and standard deviations.

	DP		MP		PP		M	
finger	mean	std	mean	std	mean	std	mean	std
I	1,00	0	-	-	1.4194	0.1540	2.1875	0.1573
II	1,00	0	1.3732	0.0819	2.3347	0.1835	3.9324	0.2925
III	1,00	0	1.5496	0.0838	2.4484	0.1359	3.5225	0.1662
IV	1,00	0	1.4234	0.0903	2.1915	0.1218	3.0532	0.1276
V	1,00	0	1.1251	0.1324	1.9009	0.1317	3.1549	0.2105

Table 5.5: Mean and standard deviation of the bone lengths extracted from 7 fine-tuned images, normalised with the corresponding bone’s DP length.

	DP		MP		PP		M	
finger	mean	std	mean	std	mean	std	mean	std
I	-	-	-	-	0.5987	0.0589	0.7810	0.0676
II	-	-	0.5815	0.0495	0.7078	0.0676	0.9191	0.0752
III	-	-	0.5774	0.0440	0.6836	0.0410	0.8896	0.0418
IV	-	-	0.5508	0.0413	0.6164	0.0482	0.7069	0.0470
V	-	-	0.5198	0.0479	0.5750	0.0396	0.7398	0.0655

Table 5.6: Mean and standard deviation of the bone head widths extracted from 7 fine-tuned images, normalised with the corresponding bone’s DP length. (Bone head widths are not calculated in the DPs.)

Based on the comparison of my table of bone length statistics with the statistics of Alexander Buryanov and Viktor Kotiuk (table 5.8), I got similar results with the proposed bone length definition, extracted from X-ray images. The largest differences are in the

	DP		MP		PP		M	
finger	mean	std	mean	std	mean	std	mean	std
I	0.5454	0.0577	-	-	0.7323	0.0359	0.7195	0.0642
II	0.5688	0.0532	0.7396	0.0486	0.9830	0.0742	1.0349	0.1268
III	0.5773	0.0355	0.7323	0.0399	0.8840	0.0527	0.7695	0.0825
IV	0.5344	0.0425	0.6689	0.0336	0.7780	0.0519	0.6916	0.0633
V	0.4856	0.0460	0.6026	0.0499	0.8238	0.0581	0.8528	0.1200

Table 5.7: Mean and standard deviation of the bone base widths extracted from 7 fine-tuned images, normalised with the corresponding bone’s DP length.

metacarpals, for which it is the hardest to define the proper segmentation.

	DP		MP		PP		M	
finger	mean	std	mean	std	mean	std	mean	std
I	0,00	-0,07	-	-	-0,04	0,01	0,05	-0,02
II	0,00	-0,14	-0,04	-0,08	-0,19	-0,13	-0,37	-0,10
III	0,00	-0,11	0,04	-0,09	-0,12	-0,08	-0,19	-0,14
IV	0,00	-0,13	-0,06	-0,10	-0,20	-0,10	-0,30	-0,16
V	0,00	-0,15	-0,01	-0,03	-0,15	-0,04	-0,21	-0,06

Table 5.8: Subtracting the table of bone lengths statistics of Alexander Buryanov and Viktor Kotiuk (table 5.8) from my results.

Chapter 6

Summary

In this study, I introduce a novel software to extract bone segmentations, length and width parameters from X-ray images supported with both automatic and manual steps, to utilize fast algorithmic computations and human anatomical knowledge, and mitigate faculties coming from attention decrease.

After studying the state of the art of hand X-ray image processing, I proposed the problems of joint localization and bone segmentation to use for parameter extraction. While there are various solutions for the tasks, these can still be improved. While machine learning-based methods have started to be part of this research field as well, until now traditional methods suppress them. It is an especially hard automatization task to produce reliable bone segmentation, based on anatomical definitions, and has no automatic solution. This may come from the lack of augmented corresponding data.

Without an automatic solution, manual segmentation is tedious work, which needs experience. Using my semiautomatic application, I was able to decrease the time spent on bone segmentation and completed the segmentation with automatic parameter extraction steps. I propose an easy-to-follow workflow of four main steps: Hand Preprocessing, Joint and Bone Localization, Bone Segmentation and Parameter Extraction. All the steps include an algorithmic solution as a baseline and an interactive way to correct them. The partial results of every step can be beneficial for other research, such as the hand or bone segmentation.

The application can be used to generate and fine-tune segmentations to create annotated data. Meanwhile, the extracted parameters can be used to parameterise a three-dimensional, generalised hand skeleton model, for personalized biomimetic hand prosthetics.

Bibliography

- [1] W. C. Roentgen, “On a New Kind of Rays,” en, *Nature*, vol. 53, no. 1369, pp. 274–276, Jan. 1896, ISSN: 0028-0836, 1476-4687. DOI: [10.1038/053274b0](https://doi.org/10.1038/053274b0). [Online]. Available: <https://www.nature.com/articles/053274b0> (visited on 10/31/2022).
- [2] I. A. Kapandji, “The physiology of the joints. Volume 1, Upper Limb,” en, *British Journal of Surgery*, vol. 57, no. 8, pp. 640–640, Dec. 2005, ISSN: 0007-1323, 1365-2168. DOI: [10.1002/bjs.1800570821](https://doi.org/10.1002/bjs.1800570821). [Online]. Available: <https://academic.oup.com/bjs/article/57/8/640/6199631> (visited on 10/29/2022).
- [3] V. K. Nanayakkara, G. Cotugno, N. Vitzilaos, D. Venetsanos, T. Nanayakkara, and M. N. Sahinkaya, “The Role of Morphology of the Thumb in Anthropomorphic Grasping: A Review,” *Frontiers in Mechanical Engineering*, vol. 3, p. 5, Jun. 2017, ISSN: 2297-3079. DOI: [10.3389/fmech.2017.00005](https://doi.org/10.3389/fmech.2017.00005). [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fmech.2017.00005/full> (visited on 11/06/2022).
- [4] B. J. Tasi, M. Koller, and G. Cserey, “Design of the Anatomically Correct, Biomechatronic Hand,” 2019, Publisher: arXiv Version Number: 1. DOI: [10.48550/ARXIV.1909.07966](https://doi.org/10.48550/ARXIV.1909.07966). [Online]. Available: <https://arxiv.org/abs/1909.07966> (visited on 10/29/2022).
- [5] D. Joshi and T. P. Singh, “A survey of fracture detection techniques in bone X-ray images,” en, *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4475–4517, Aug. 2020, ISSN: 0269-2821, 1573-7462. DOI: [10.1007/s10462-019-09799-0](https://doi.org/10.1007/s10462-019-09799-0). [Online]. Available: <http://link.springer.com/10.1007/s10462-019-09799-0> (visited on 10/31/2022).
- [6] S. Simu, S. Lal, K. Fadte, and A. Harlapur, “Fully automatic segmentation of phalanges from hand radiographs for bone age assessment,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 7, no. 1,

- pp. 59–87, Jan. 2019, Publisher: Taylor & Francis _eprint: <https://doi.org/10.1080/21681163.2017.1416491> ISSN: 2168-1163. DOI: 10.1080/21681163.2017.1416491. [Online]. Available: <https://doi.org/10.1080/21681163.2017.1416491> (visited on 11/29/2023).
- [7] X. Pan, Y. Zhao, H. Chen, D. Wei, C. Zhao, and Z. Wei, “Fully Automated Bone Age Assessment on Large-Scale Hand X-Ray Dataset,” en, *International Journal of Biomedical Imaging*, vol. 2020, e8460493, Mar. 2020, Publisher: Hindawi, ISSN: 1687-4188. DOI: 10.1155/2020/8460493. [Online]. Available: <https://www.hindawi.com/journals/ijbi/2020/8460493/> (visited on 11/29/2023).
- [8] D. K. Eng, N. B. Khandwala, J. Long, *et al.*, “Artificial Intelligence Algorithm Improves Radiologist Performance in Skeletal Age Assessment: A Prospective Multicenter Randomized Controlled Trial,” *Radiology*, vol. 301, no. 3, pp. 692–699, Dec. 2021, Publisher: Radiological Society of North America, ISSN: 0033-8419. DOI: 10.1148/radiol.2021204021. [Online]. Available: <https://pubs.rsna.org/doi/10.1148/radiol.2021204021> (visited on 06/03/2024).
- [9] D. D. Martin, A. D. Calder, M. B. Ranke, G. Binder, and H. H. Thodberg, “Accuracy and self-validation of automated bone age determination,” en, *Scientific Reports*, vol. 12, no. 1, p. 6388, Dec. 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-022-10292-y. [Online]. Available: <https://www.nature.com/articles/s41598-022-10292-y> (visited on 10/31/2022).
- [10] BAA, *Rсна dataset*, <https://universe.roboflow.com/baa/rsna-c7v1m>, Open Source Dataset, visited on 2024-06-06, Jan. 2024. [Online]. Available: <https://universe.roboflow.com/baa/rsna-c7v1m>.
- [11] Yolo, *Hand-bones dataset*, <https://universe.roboflow.com/yolo-qdpfg/hand-bones-2raxr>, Open Source Dataset, visited on 2024-06-06, Sep. 2023. [Online]. Available: <https://universe.roboflow.com/yolo-qdpfg/hand-bones-2raxr>.
- [12] U. Malaya, *“hand bones” dataset*, <https://universe.roboflow.com/universiti-malaya-tchmy/-hand-bones>, Open Source Dataset, visited on 2024-06-06, Oct. 2023. [Online]. Available: <https://universe.roboflow.com/universiti-malaya-tchmy/-hand-bones>.
- [13] G.-S. Ham and K. Oh, “Learning Spatial Configuration Feature for Landmark Localization in Hand X-rays,” en, *Electronics*, vol. 12, no. 19, p. 4038, Jan. 2023,

Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-9292. DOI: [10.3390/electronics12194038](https://doi.org/10.3390/electronics12194038). [Online]. Available: <https://www.mdpi.com/2079-9292/12/19/4038> (visited on 11/30/2023).

- [14] A. Bielecki, M. Korkosz, and B. Zieliński, “Hand radiographs preprocessing, image representation in the finger regions and joint space width measurements for image interpretation,” en, *Pattern Recognition*, vol. 41, no. 12, pp. 3786–3798, Dec. 2008, ISSN: 00313203. DOI: [10.1016/j.patcog.2008.05.032](https://doi.org/10.1016/j.patcog.2008.05.032). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0031320308002094> (visited on 11/08/2022).
- [15] A. Yuksel and T. Olmez, “Automatic Segmentation of Bone Tissue in X-Ray Hand Images,” in *Adaptive and Natural Computing Algorithms*, M. Kolehmainen, P. Toivanen, and B. Beliczynski, Eds., vol. 5495, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 590–599, ISBN: 978-3-642-04920-0 978-3-642-04921-7. DOI: [10.1007/978-3-642-04921-7_60](https://doi.org/10.1007/978-3-642-04921-7_60). [Online]. Available: http://link.springer.com/10.1007/978-3-642-04921-7_60 (visited on 10/31/2022).
- [16] Y. Gong, *Decompose X-ray Images for Bone and Soft Tissue*, arXiv:2007.14510 [cs, eess], Jul. 2020. [Online]. Available: <http://arxiv.org/abs/2007.14510> (visited on 11/21/2022).
- [17] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, *et al.*, “The RSNA Pediatric Bone Age Machine Learning Challenge,” *Radiology*, vol. 290, no. 2, pp. 498–503, Feb. 2019, Publisher: Radiological Society of North America, ISSN: 0033-8419. DOI: [10.1148/radiol.2018180736](https://doi.org/10.1148/radiol.2018180736). [Online]. Available: <https://pubs.rsna.org/doi/10.1148/radiol.2018180736> (visited on 10/29/2022).
- [18] J. M. Sotoca, J. M. Iñesta, and M. A. Belmonte, “Hand bone segmentation in radioabsorptiometry images for computerised bone mass assessment,” *Computerized Medical Imaging and Graphics*, vol. 27, no. 6, pp. 459–467, Nov. 2003, ISSN: 0895-6111. DOI: [10.1016/S0895-6111\(03\)00053-3](https://doi.org/10.1016/S0895-6111(03)00053-3). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0895611103000533> (visited on 11/29/2023).
- [19] B. I. Formanek, “Anatómiai morfológiai paraméterek algoritmikus kinyerése az emberi kézről alkotott orvostechnológiai képekből, biomimetikus protézisfejlesztési célokra,” 2022.

- [20] A. Buryanov and V. Kotiuk, "Proportions of Hand Segments," en, *International Journal of Morphology*, vol. 28, no. 3, Sep. 2010, ISSN: 0717-9502. DOI: 10.4067/S0717-95022010000300015. [Online]. Available: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0717-95022010000300015&lng=en&nrm=iso&tlng=en (visited on 11/02/2022).
- [21] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979, ISSN: 0018-9472, 2168-2909. DOI: 10.1109/TSMC.1979.4310076. [Online]. Available: <http://ieeexplore.ieee.org/document/4310076/> (visited on 11/10/2022).
- [22] R. C. Gonzalez and R. E. Woods, *Digital image processing*. New York, NY: Pearson, 2018, ISBN: 978-0-13-335672-4.
- [23] E. Zemmour, P. Kurtser, and Y. Edan, "Automatic Parameter Tuning for Adaptive Thresholding in Fruit Detection," *Sensors*, vol. 19, p. 2130, May 2019. DOI: 10.3390/s19092130.