University College London - Department of Mechanical Engineering

Image Analysis Methods for Dental Orthodontic Diagnosis

Student:
German Malsagov

Supervisor:
Dr. Adam Wojcik

Word Count: 6944

April 26, 2017

I, German Malsagov, confirm that the work presented in this report is my own. Where information has been derived from other sources, I confirm that this has been indicated in the report.

Contents

*

List of Figures

List of Tables

Nomenclature

$\hat{y}$	Height of the Gap Valley
μ	Mean Value of Statistical Set
σ	Standard Deviation of Statistical Set
R_{i}	Boundary of the Region of Interest
v_i	Horizontal Integral Projection
a	Output of Artificial Neural Networks
ALT	Adaptive Local Thresholding
ANN	Artificial Neural Networks
CT	Computed Tomography
ED	Edge Detection
GV	Gap Valley
h	Height of an Image
I(i,j)	Pixel Intensity at Row j Column i
MRI	Magnetic Resonance Imaging
ROI	Region of Interest
w	Width of an Image
x$_($c)	Coordinate of Image Centre on x-axis
y$_($c)	Coordinate of Image Centre on y-axis

Abstract

This project aimed to develop software that could analyse 2-dimensional X-Ray images in order to assist orthodontists with treating malocclusions (teeth misalignment). This work presents two methods that were developed in order to measure angular position of teeth on the supplied images. The work completed thus far allows to obtain measurements of teeth angles of high accuracy with minimum human intervention.

The first model is based on application of pattern recognition and machine learning techniques. Prior to this step, panoramic image is pre-processed to remove excessive noise around edges of teeth and improve contrast levels to highlight teeth and suppress background structures. Once the quality of image is improved, the boundaries of teeth are extracted using edge detection algorithm developed on the basis of Sobel filter. The obtained contours are fed into Artificial Neural Networks for analysis of similarities with the training database. The result of this step is either a recognized tooth shape, which could be used for measurements, or a rejected pattern. However, this method did not produce sufficient results. Thus, alternative approach was taken to solve the issue.

A novel method was designed to obtain angular position of teeth. Second algorithm involves evaluation of intensity profile of panoramic radiograph, which includes calculation of the gap valley (line separating upper and lower jaws). Having cut the image into 21 equally-sized strips, the sum of the intensities along each row of respective strip is found. The smallest value corresponds to the line with lowest overall brightness, thus, is assumed to lie within the gap valley. Polynomial approximation, applied to a set of coordinates of obtained lines, is used to find an approximate curve passing through the gap valley. This curvature is then translated up and down and intensity distribution is analysed similarly in order to find the position of gaps between necks of teeth. These allow to separate individual teeth by drawing line from the gap valley to the end of image through neck gaps. The tooth angles are determined by averaging the angles of two lines, which bounded a single tooth. Comparison of this method with manual measurements showed that the results were accurate within 7.9%. In addition, the algorithm measured angles automatically within 7 seconds, which satisfies the required runtime.

Key Words: Computer Vision, Intelligent Segmentation, Pattern Recognition, Orthodontics, Dental Diagnosis.

Acknowledgements

I would like to express my sincere gratitude towards my project supervisor, Dr Adam Wojcik, for his continuous support throughout this project. I believe that without his guidance I would not be able to complete this work.

1 Introduction

Malocclusion is an incorrect alignment of the teeth in maxilla (upper jaw) and mandible (lower jaw) when they approach each other as the jaw closes. Orthodontists utilise bracers and retainers to apply forces onto teeth in order to move them to their ideal positions. At present, there is no quantitative method to model the reaction of teeth. In fact, dentists largely rely on their experience when designing and adjusting individual treatment plan. As a result, the efficiency of therapy is often determined by the level of expertise of orthodontic practitioners. Thus, a technique to quantify the response of the teeth due to applied forces is sought in order to enable them to predict changes in teeth position throughout treatment process. This has potential to drastically optimize orthodontic procedure. The simplest way of tracking the displacement of teeth is by means of medical imaging techniques because they are relatively quick to perform and allow to reveal the size and shape of internal structures without direct contact with patients.

This work is going to focus on analysing medical images of jaw structures in order to extract angular data of corresponding teeth. It was decided to consider 2-dimensional panoramic radiographs only due to wide prevalence of this particular scanning method and hence large availability of images online. On the other hand, panoramic x-rays are often of low quality and possess random noise due to errors in signal processing, which originates from limitations of the technique. As a result, the images have to be pre-processed prior to applying the algorithm in order to improve their quality and account for these errors. Besides, there are other advantages and disadvantages of the mentioned technique, which will be discussed in the following sections of the report. The research will evaluate two methods that were developed in order to achieve the set of objectives. The first technique makes use of pattern recognition and machine learning algorithm with several pre-processing stages such as contrast enhancement, edge detection, noise removal and others. The second method is based on intelligent segmentation process, which aims to partition image into several regions of interest each containing one tooth at maximum. After that, an algorithm computing gradients of the line separating teeth is applied in order to extract angular information corresponding to each tooth. The report also provides background information on orthodontics as well as the description of the above-mentioned methods with the analysis of the results and future work explained at the end.

2 Global Summary

2.1 Aims and Objectives

There are a number of desired outcomes this project aims to achieve. In general, the primary objective of this project is to successfully produce an image-based system for orthodontic treatment diagnosis. More specifically, this will represent a program capable of analysing 2-dimensional x-ray images to accurately determine their angular parameters. Secondly, the software should operate with minimum human intervention i.e. be as automated as possible. In addition, it is essential that the system is tested, is fully functional and meets all the constraints listed in the following section. Lastly, it should be supported by most computers, in other words, be developed and optimized for a widely-used programming language.

2.2 Design Specifications

The user of the developed product will be interacting with the system to help measure the angles of the teeth. The software allows the user to extract required information by completing series of computational operations to analyse panoramic radiographs provided by the user. The product can run on all PC's supporting Matlab version R2016a. Detailed system requirements for this version could be found in Appendix A.

2.3 Design Criteria

The success of this project depends primarily on meeting the objectives outlined in the previous part. It enables to make summary of the main goals and results this project aims to achieve. However, it is important to evaluate how well the solution has met the requirements set initially. For this, a list of success criteria was designed to evaluate each key step of the project. All criteria must be met in order to consider project to be a success. These criteria were listed below:

- Image is partitioned into individual teeth
- All teeth contained within the image are identified
- The boundaries of segmented teeth remain integral
- The system is capable of producing accurate measurements of teeth angles
- Required information is obtained by selecting an image and running script
- Software is simple to navigate in
- Results are produced within 5-minute interval

2.4 Structure of the Report

- Chapter 3 - gives background information about current issues related to orthodontia and presents justification of selected imaging technique.
- Chapter 4 - provides an overview of two methods that were developed and tested in order to achieve the objectives of this work.
- Chapter 5 - presents the results obtained using the proposed method as well as their analysis and several improvements that could be done.

3 Literature Review

A comprehensive literature review was conducted to accomplish designing of the solution for orthodontic diagnosis. In this section, the main issues and limitations for the work will be defined as well as approaches to overcome them will be explained. In addition, this section will include some information regarding the selection of imaging technique, which essentially determined the direction of this work.

3.1 Teeth Misalignment

Teeth imperfection is a common issue that people face as they grow up. According to NHS (2014), more than 30% of children at the age of 12 have dental issues related to orthodontia in the UK. Most humans do not possess ideal smile but slight deviation in their teeth arrangement does not affect their life significantly. However, high degree of malocclusion can make eating process difficult and cause issues with sleeping (Abanto et al., 2010) and pain due to strains in the jaw muscles (Rusanen et al., 2011). Moreover, complicated cases of teeth misalignment can effectively change the facial features of the person. Visiting orthodontist could become a solution to the most of these problems.

3.2 Orthodontic Treatment

Orthodontic treatment is aimed to improve the appearance by correcting the position and function of the crooked or twisted teeth. Treatment planning involves accurate analysis of patient's teeth arches via calculating several dimensions including tooth position, shape and orientation with respect to a specific coordinate system. Based on this analysis an ideal teeth arch is designed, which will then be used to assess the result of the treatment. Movement of teeth to the desired positions is achieved by application of forces onto each tooth individually. Currently, the most common technique is to place special bracers on every tooth and connect them by means of a wire that drags them towards final position due to tension force.

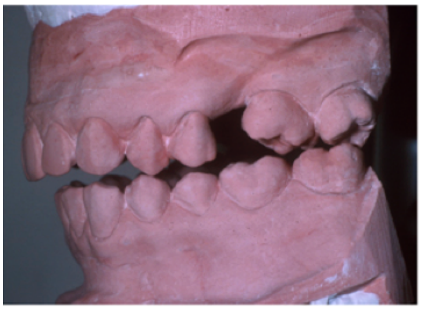


Figure 1: Representation of dental cast (Cordray, 2016).

3.3 Disadvantages of Current Methods

Orthodontists largely rely on their knowledge and common sense when predicting the movement of a particular tooth. To track the progress a method of extracting precise angular data is required. Currently, it is mostly implemented by physical measurements of the produced dental casts. One of the disadvantages of the described method is that it allows to examine only crowns, while evaluation of roots is equally important (Harrell, Hatcher and Bolt, 2002). Furthermore, in the case of hidden teeth being present, dentition cast would not provide sufficient information to design optimum treatment strategy. A human error associated with measurements cannot be disregarded either. Lastly, the process of fabricating dental casts is not perfect in terms of patient's comfort. It requires a person to keep jelly-like substance in mouth for a period of time. Some humans can't tolerate this due to sensitive gag reflex. Hence, an alternative, more comfortable way of measuring teeth movement is desired to ease the procedure for both the patient and the dentist.

3.4 Justification of Selected Imaging Technique

Automated diagnostic imaging has become one of the most important branches of picture processing recently. The goal of this field is to represent the internal body parts visually for medical analysis. There are various types of imaging techniques for body examination, which include X-Ray, CT (Computed Tomography) and MRI scans (Magnetic Resonance Imaging), ultrasound and others. Nevertheless, both MRI and CT appear to be too expensive and time consuming to be used in a routine examination procedures. In addition, it is critical to consider the fact that most hospitals and private dental clinics do not possess advanced diagnosis equipment. In this respect, it would be beneficial to develop a solution that could be used by as many dental practices as possible. Hence, the program should utilize images produced by means of widely applied method of diagnosis. According to Rushton, Horner and Worthington (2001), the analysis of forty-one general dental practices showed that in 57.1% of cases patients did not take any radiograph apart from panoramic. Based the above, panoramic radiography provides the best trade-off between price, time efficiency and prevalence. Consequently, it was selected as an image source for this project.

3.5 Panoramic Radiograph

Panoramic radiograph is a dental x-ray, which provides a 2-dimensional representation of upper and lower jaws by panoramically scanning patients from ear to ear. Figure 2 illustrates a typical x-ray machine used for taking panoramic dental radiographs. The patients rest their chin on the central mounting and bites a mouth opener mouthpiece (Budinger and Brahme, 2014). Once the patient's mouth is fixed in the central position and the height is adjusted in accordance with anthropometrics of the person, the machine starts to rotate around his/her head making a semi-circle from ear to each. The source, located inside the arm, emits x-ray beams, which, passing through the jaws, project their panoramic scan on the opposite arm containing x-ray film or detector.

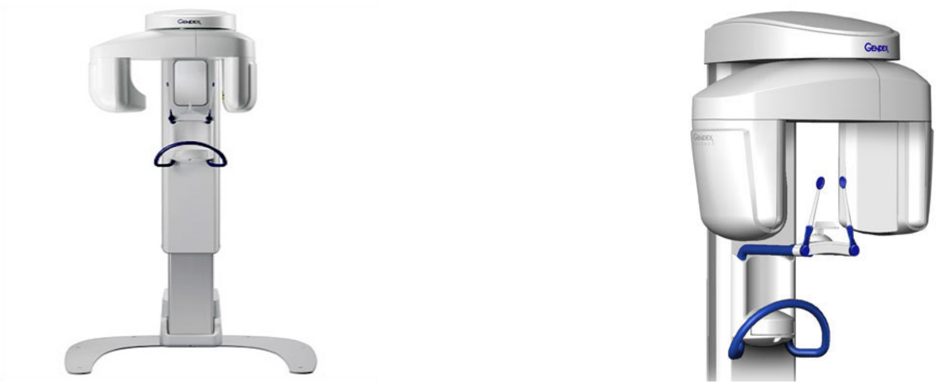


Figure 2: Typical panoramic dental x-ray machine (RenewDigital.com, 2017).

3.6 Approach

Automatic object detection was successfully applied to many fields of image processing including face recognition, navigation of autonomous vehicles, number plates detection and many others. As a result, classification of teeth according to their shape seemed as a promising method to be utilized in orthodontic diagnosis. The developed algorithm will be presented in the next section of the report.

Since the objective of the work was to obtain the dimensional data of individual teeth, a solution capable of separating original radiographs into small regions containing one tooth at maximum was required. Thus, alternative method was developed to accomplish this goal. The description of the proposed algorithm will be presented in Chapter 5 of this report.

4 Methodologies

In this chapter two models developed for identification of individual teeth will be described. Firstly, it will identify the main characteristics of the object detection system as well as describe interactions between successive steps. Moreover, second method based on intelligent segmentation will be explained further. Lastly, this chapter will provide the rationale behind several concepts that were implemented in both designs and will form solid foundation for the analysis of the results and future work.

4.1 Object Recognition Method

This section is going to describe the main steps of the developed model. As was established earlier in the report, teeth recognition was a primary task for this design project. Due to variation in teeth shapes and orientation, the set of target data was required to train an algorithm to recognise patterns. A function to calculate the angles of teeth would be built after the primary goal is achieved. Several pattern recognition models were studied including statistical and syntactic approaches, template matching and neural networks. Although, all of these systems were capable of producing the desired result, it was decided to select Artificial Neural Network (ANN) due to its relative concept simplicity (compared to other techniques) and ability to learn from self-experience (Kpalma and Ronsin, 2007). The sequence of steps performed to recognize tooth could be seen on Figure 3. Most notable features as well as description of key steps will be provided further.

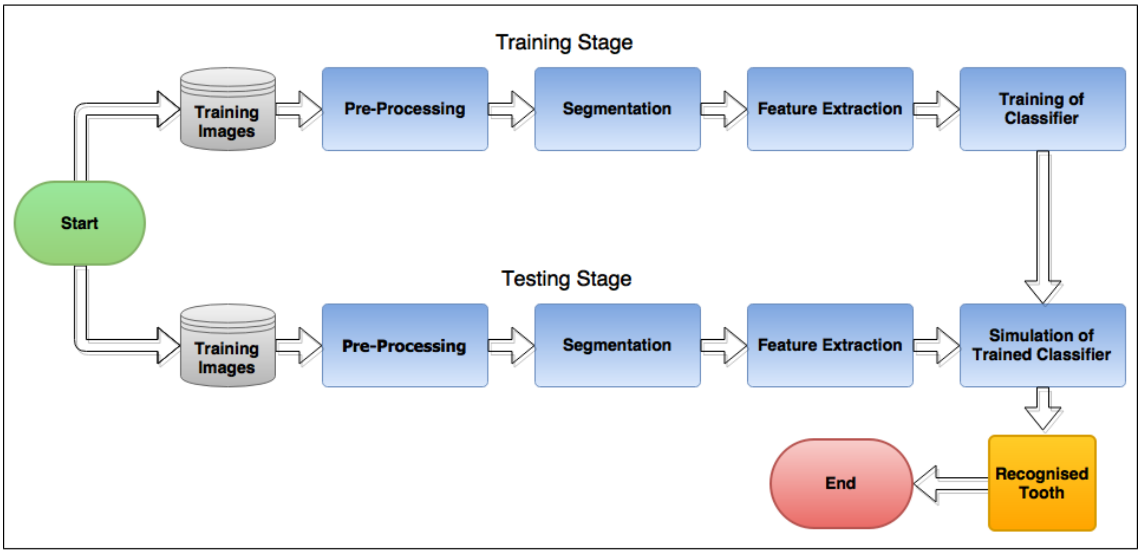


Figure 3: Teeth recognition model based on system developed by Prasad et al. (2013)

4.1.1 Definition of Key Features

Automatic measurement of the dimensions and inclination of the teeth could only be implemented if the program is capable of distinguishing the teeth from the background. Thus, an accurate teeth recognition algorithm is key to solve this issue. The shape of the teeth makes them distinct from the background structures. In essence, there are four tooth types present in regular human mouth: incisors, canines, premolars and molars (Hobson, McCabe and Hogg, 2001). Incisors and canines as well as premolars and molars have similar shape, which makes the analysis easier as only two types of shapes are expected. However, their identification on input images solely by virtue of pattern recognition techniques is strictly limited due to following limitations:

- Low resolution, magnification and distortion are sources of noise, which make the analysis more complicated. Consequently, image filtering may not reveal the exact boundaries of the teeth. Hence, the code should also be able to predict where the tooth contour should go.
- Complex irregularities such as twisted, shifted and overlapping teeth will constitute more sophisticated shapes.
- Teeth can vary in size, shape and orientation for different people. Thus, the model should account for these differences.
- There is no exact equation, which can define the form of the tooth. Each tooth is formed throughout life and can have deviations as well as broken and restored parts. Thus, the rules need to be established gradually through evaluation of sample data and learning.

An analogy with human brain could be drawn in this case. People often see familiar patterns in completely different situations (star constellations, cloud shapes). This is because human brains can identify different shapes irrespective of minor visual variation or discontinuity in shape characteristics. In addition, it learns particular features based on visual experience. It is also worth noting that human brain does not rely on conditions set initially such as aspect ratio, dimensions and orientation. These points could be applied to the teeth recognition problem. In order to measure the dimensions and orientation, the program should be able to recognize the shape of the tooth in spite of noise and irregularities present on image scans. For this to occur, the problem should be approached by a combination of pattern recognition and machine learning fields, which often considered inseparable in image processing area. Machine learning algorithm would simulate human brain's ability to learn from experience and pattern recognition would provide visual information for it. These areas of image processing field will be reviewed in the following sections of this report.

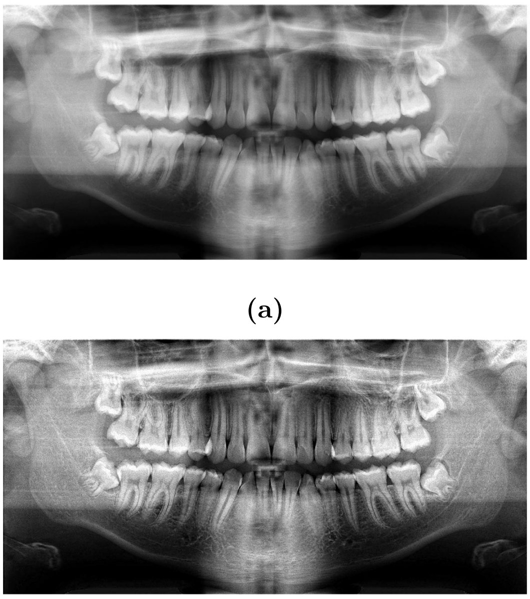
4.1.2 Pre-Processing

Pre-processing is a set of enhancement procedures performed on the original image to assure that the assumptions of the described method are satisfied. Successful identification of objects requires the image to pass several pre-processing stages primarily to reduce the amount of noise present on the radiograph. Accomplishment of this stage is done before identification method can be applied to the image in order to extract useful information. The list of pre-processing steps, which were completed during both training and recognition stages are shown below:

Grayscale conversion – colour image was represented in shades of grey. The main interest in the radiographs are the boundaries of the teeth, analysis of which does not require colour properties. Hence, colour elimination was conducted in order to narrow pixel data down to brightness information only. Moreover, grayscale images could be visualized in 3D in terms of two spatial dimensions and one intensity dimension allowing to observe intensity gradients as well as peaks and valleys.

Noise reduction – filtering and smoothening to decrease noise caused by the errors in image acquisition process. Certain level of blurriness could be observed around teeth roots (see Figure 3). In addition, random noise could be seen around the image. Noise reduction technique was applied to ensure wrong information is not introduced to the system and hence, improved the outcome of edge detection process. Several methods were tested in order to reduce the degree of blurriness of panoramic radiographs. Wiener filter performed better in image restoration than Gaussian and other smoothening filters. This noise reduction filter enables to recover blurred image and smoothen noise simultaneously by applying a parameter which compromises both distortion-correction and noise-reduction effects (Chen et al., 2006).

Contrast enhancement – improving image sharpness in order to make objects more distinguishable from the adjacent structures. It ensures that relevant information is scaled to simplify the detection process. In essence, Top-and-Bottom hat filtering technique was applied in attempt to make teeth boundaries more distinct and reduce excessive noise. This method is based on analysis of intensity plot throughout image. Pixels with high brightness correspond to peaks on the graph, while lower intensity areas are represented as valleys. Top-hat filter enlarges peaks by few pixels improving overall contrast of an image, whereas Bottom-hat emphasizes dark regions of an image. Significant contrast enhancement was achieved by subtracting two filtered images. The resulting image had darker regions eliminated thus, leaving peak intensities only (Figure 4b).



(a)

(b)

Figure 4: Original image (a) (DrDerhamDental.com, 2017) and resulting image (b) with contrast enhancement applied using author's methodology.

4.1.3 Segmentation

Image segmentation is one of the most difficult tasks in image processing. It plays vital role in subsequent steps of analysis. Segmentation simplifies the analysis by drastically reducing the amount of data that needs to be processed. This reduction is achieved via partitioning original image into a number of small regions according to their pixel parameters such as spatial frequency, intensity and average grey level (El Joumani et al., 2017). The goal of this step is to isolate objects, which fit into the criteria to allow classifier to extract features. They also aim to reduce the amount of data, which needed to be processed making classification simpler. As there are numerous methods that could be used, two image treatment techniques were tested, and the one that provided the best results was selected. The methods tested were *Adaptive Local Thresholding (ALT)* and *Edge Detection (ED).*

Adaptive Local Thresholding algorithm separates objects from background in an image with non-uniform illumination. Having estimated local threshold value of each pixel, the image is converted into black and white cells accordingly (Sezgin and Sankur, 2004). By examining Figure 5, it could be observed that ALT failed to threshold foreground pixels from background in the region of interest near tooth roots due to significant noise around them. In addition, presence of cervical spine made central incisors barely distinguishable from the background. This issue was caused by the procedure of taking panoramic radiographs (refer to Chapter 2). Overall, ALT appeared to be too sensitive to noise and have under/over-segmentation tendency. Thus, it was considered insufficient to provide optimum results for further teeth recognition. Consequently, several edge detection methods were tested.

Edge Detection is a process that allows to discontinuities in a digital image corresponding to boundaries between two adjacent regions having different pixel intensities. Thus, having arranged these sets of pixels into line segments one would obtain boundaries of objects within the image (Rani and Kumari, 2016). This step will help to extract useful information about objects of interest (teeth boundaries) as well as significantly reduce the amount of processing data. In case of panoramic radiographs, teeth have higher brightness than surroundings. Consequently, two edge filters (Sobel and Canny) were employed and their results were compared respectively. The Sobel operator produced the best results (see Figure 6), which could be explained by two reasons. Firstly, it utilizes averaging techniques, thereby smoothening the boundaries and reducing the effect of surrounding noise. Also, compared to Canny, it not only takes into account the magnitude of borders, but their direction as well. This is achieved by multiplying each pixel of an image by two 3x3 kernels (representing horizontal and vertical edges) in order to determine their gradients. The angle corresponds to either level or vertical edge depending on the highest result (Gao et al., 2010). Hence, the designed edge detection algorithm was based on Sobel filter (refer to Appendix B).

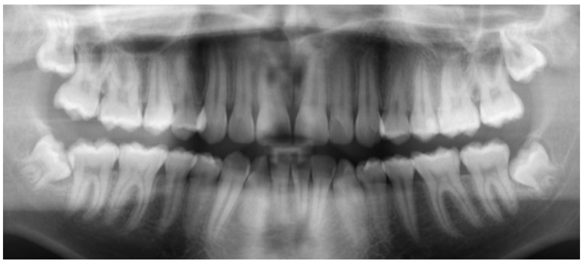


Figure 5: The panoramic dental x-ray of a patient (DrDerhamDental.com, 2017).

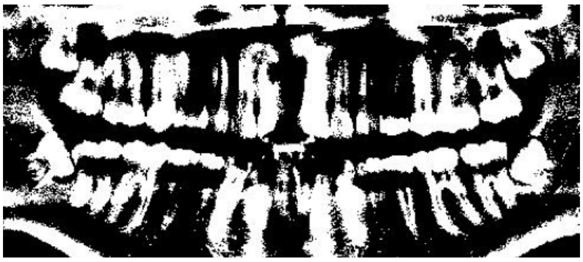


Figure 6: The resulting image after applying Adaptive Thresholding method used by the author.

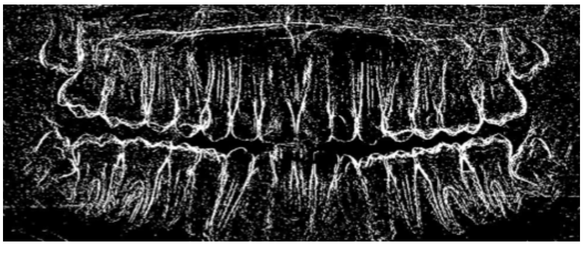


Figure 7: The resulting image after completing edge detection via Sobel filter method developed by the author.

4.1.4 Feature Extraction

Once the teeth edges were identified, their corresponding pixels need to be stored in arrays for further analysis. Feature extraction stage identifies the most notable features of the tooth such as edges and records pixel intensities corresponding to them. These values will be stored in the special list in a matrix form. The produced list would be used for subsequent training and classification steps.

4.1.5 Training and Classification

Training and classification is the stage where the neural network (ANN) learns how to make a decision about particular pattern data. The shape that needs to be recognized is matched against the stored data generated during preliminary stages. In addition, fire rules are set in the function to allow processing of the information that is not included in the training data set. Thus, database of 20 molars was created for training stage (Appendix C). The contours of individual teeth were cut out from various panoramic radiographs by means of intelligent scissors tool. These images were used to teach ANN to recognise teeth.

Artificial Neural Networks

Artificial Neural Network could be represented as a large set of simple processing elements (neurons) highly interconnected to each other. These neurons (shown on Figure 6) work in parallel to solve particular tasks. ANN could be trained to execute particular function by adjusting these connections (weights) between neurons. The weights determine the significance of each neuron's contribution The system requires target data to compare with input values in order to match the output (a) with target. This means that ANN is able to learn based on the example data. Learning is achieved by operating neurons in training mode when a selection of training patterns is fed into the system. This collection consists of two types of data: positive and negative set of patterns corresponding to 1's and 0's respectively. The neurons are taught to fire for specific 1-patterns and not to for 0's. Once ANN is trained, if the unknown pattern is introduced to neurons, the firing rule will be applied to classify this type of data. Firing rule are set to explain neurons when they should fire. Fundamentally, these rules allow to compare input data with ones and zeroes. If pattern has enough similar elements with the positive training set, neuron fires.

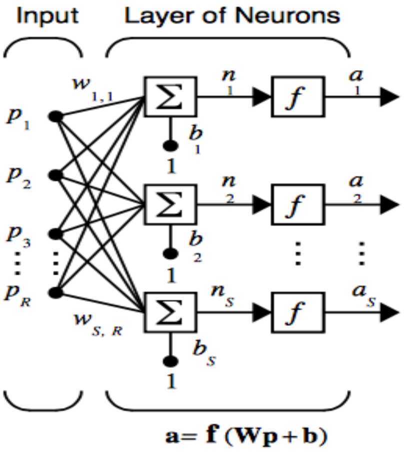


Figure 8: Representation of Simple Neuron Network (Demuth and Beale, 2002).

Nevertheless, application of pattern recognition and machine learning techniques to dental imaging problem did not produce sufficient results. This primarily has to do with the necessity to analyze highly complex databases with a huge number of possible variables. Parameter variations throughout boundaries were caused by excessive noise around the edges of the teeth. Moreover, the size of statistical model was not large enough to cover the majority of possible dimensions and orientations of teeth. Effective object recognition typically requires training dataset to have thousands of samples (Papa et al., 2012). Also, pattern recognition techniques require high performance computing, which often is not available, thereby, violating the initial objective of producing a solution accessible for the majority of dental practices. Lastly, there were some issues with efficiency, since the speed of neural networks depended upon the complexity of the processed data.

As a result, the decision to develop alternative solution was made. The design of this model will be described in the next section of the report.

4.2 Method Based on Intelligent Segmentation

This section is going to present an alternative solution developed in order to solve current issue. This method was designed from the first principles of computer vision as the first model failed to produce sufficient results due to large amount of data to be processed. Thus, the ultimate goal was to break down complex panoramic radiograph into single teeth by defining region containing all teeth, locating the gap between jaws and separating individual teeth by determining position of gaps between necks of teeth. The flow diagram of this method could be seen on Figure 7.

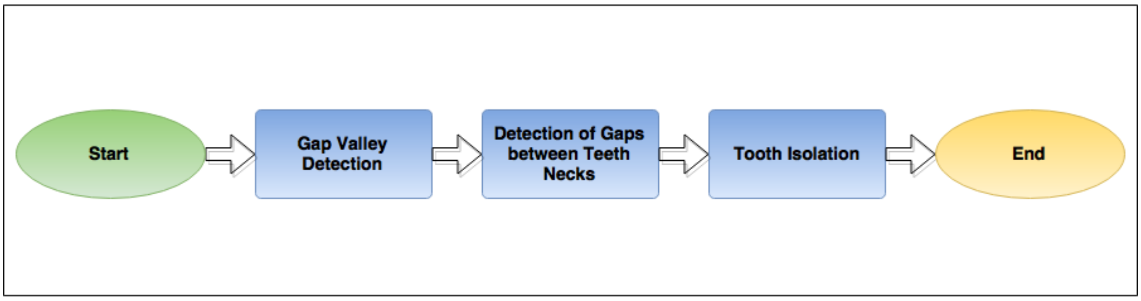
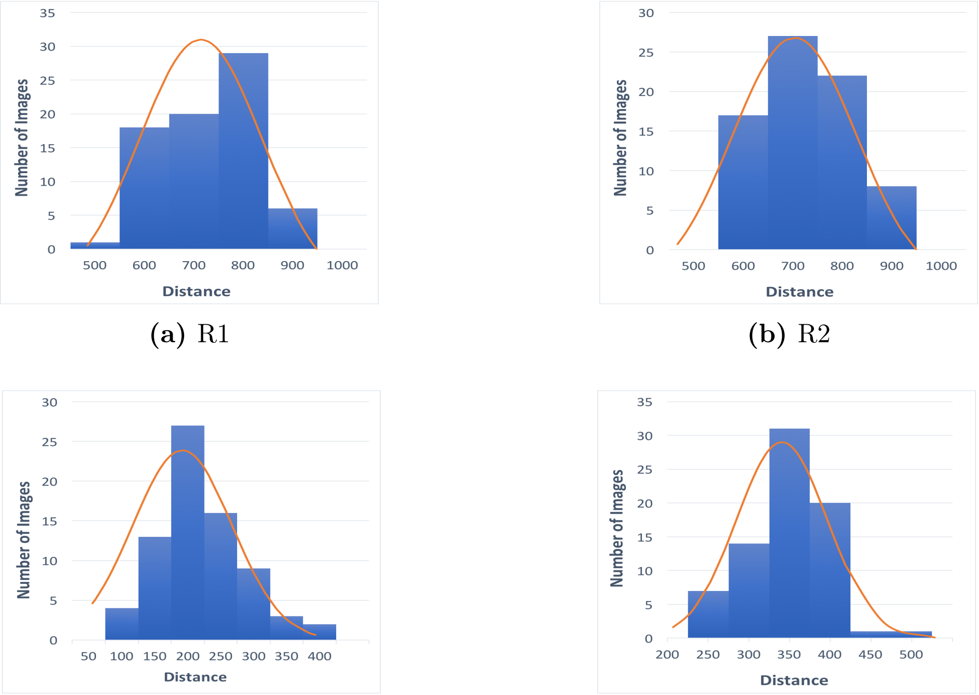


Figure 9: Flow diagram of the integral projection method.

4.2.1 Region of Interest Definition

In order to proceed, it was required to define the Region of Interest (ROI) that most expected to contain all teeth present on the image. In essence, there are three regions present on dental X-Ray images: the background, mainly represented by soft tissues, which has the lowest intensity. The bone structures, having moderate intensity levels and the teeth, which have the highest intensity among these three. Nevertheless, bones have similar intensity to the teeth in some images, which could result into incorrect segmentation (Rad et al., 2015). Consequently, this step was accomplished in order to reduce the number of errors in partitioning algorithm caused by intensity distortion from adjacent structures. The boundaries of ROI were determined based on statistical analysis of the database composed of 74 panoramic images of random people, which were collected from the internet. The main goal was to crop the region containing all the teeth and remove the maximum amount of noise. As all images were obtained from different sources, their dimensions varied significantly making analysis difficult. Therefore, the images were stretched to the standard size of 2400 x 1200 pixels by means of resize function. Four distances (R_1, R_2, R_3, R_4) were measured from the center point $[(x_c, y_c)=w/2=1200, h/2=600]$ for each image in the database and recorded into a table (refer to Appendix D).

Having obtained all sets of dimensions, four histograms were produced (shown on Figure 10). The smooth lines correspond to the approximate normal distributions obtained using means (μ) and standard deviations (σ) of each statistical set. Using these curves the appropriate value for each R_i was determined. Corresponding distances were found by applying 95% confidence interval. The calculated values are: R_1=875, R_2=850, R_3=318, R_4=428, which formed ROI that could be seen on Figure 11. This particular model allows to extract the statistically-defined margins and ensures that certain variations in jaw size and positioning during scanning will not affect the teeth recognition process significantly. In fact, jaws of smaller size will be cropped effectively with a narrow region of non-useful information. However, the 95% confidence level leads to oversized images being partially cropped meaning that teeth on the edges would not be recognized, but probability of this was determined to be about 5%, which means the accuracy of algorithm is acceptable. Meanwhile, the images fitting within defined boundaries might have larger presence of noise due to jaw size variation making it difficult to establish the ideal size for all patients.



(a) R1

(b) R2

(c) R3

(d) R4

Figure 10: Histograms for Ri (distance) values with approximate normal distribution superimposed above.

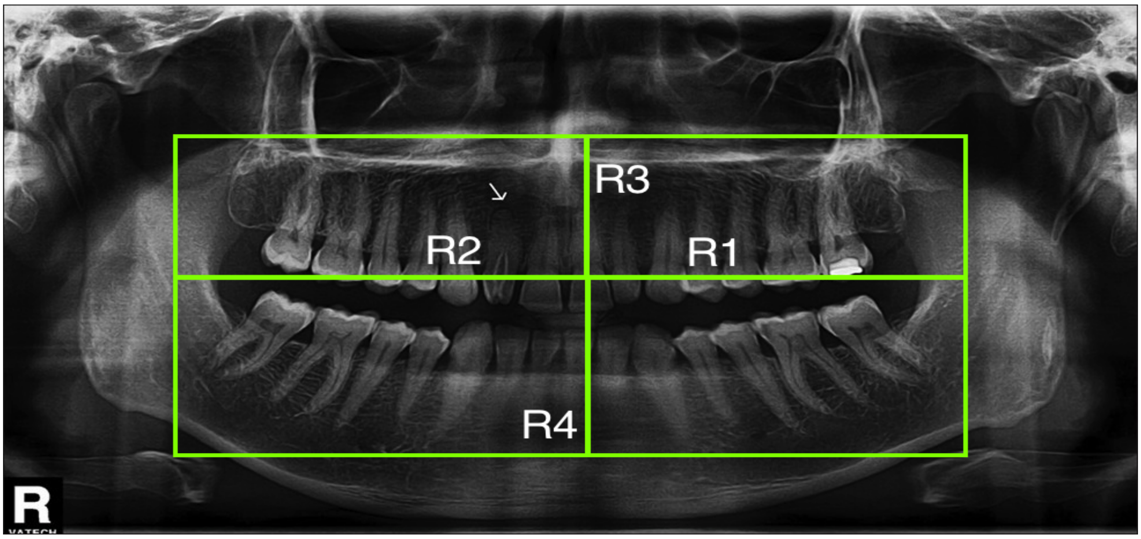


Figure 11: Four distances forming Region of Interest on patient's radiograph.

4.2.2 Separation of the Upper and Lower Jaws

During this stage, the line separating the upper and lower jaws is determined. To produce panoramic radiograph, patients are required to bite a mouth opener mouthpiece, thereby leaving a gap between jaws. This gap is presented on the image as a dark horizontal stripe. This stripe will form a valley on the intensity histogram in the y-axis (FIGURE), which will be called gap valley (GV) from now on. In order to separate maxilla and mandible, the valley's set of coordinates must be obtained. Thus, an algorithm that could compute the position of the gap was written based on following assumptions:

- The gap between upper and lower jaws is expected to have the lowest integral intensity throughout the image
- The slope of the gap valley does not vary significantly

ROI was divided into vertical stripes, each equal to the average width of the tooth. The stripes consisted of several pixels located along horizontal line, the length of which was equal to the height of ROI. The sum of the intensities along each row represented horizontal integral projection of corresponding row. Based on our assumptions, the height of the gap valley ($\hat{y}$) corresponded to the lowest horizontal integral projection of the stripe. Thus, pixel intensities along each row were summed up and the minima was found in order to locate the position of the gap valley. The value of horizontal projection was determined as:

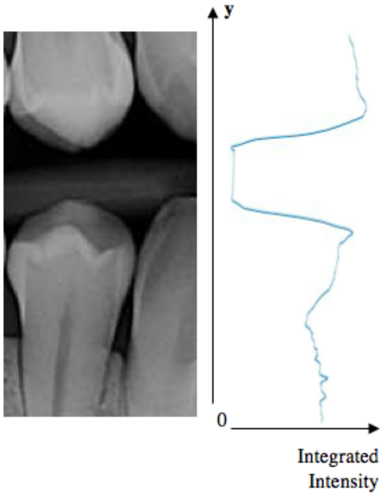


Figure 12: Integral projection on y-axis

$$v_i = \sum_{i,j=1}^{w} I(i,j)$$

where, I(i,j) represents the intensity of a pixel at row j and column i.

As was established earlier, ROI may sometimes capture non-useful information on the sides. In effect, this means that analysis of the edges could not be performed at the beginning as gap valley may not be present on them. As a result, it was decided to initiate the algorithm from the centre of the image and continue towards both left and right sides. Initial point of separation was found by inspecting the central stripe and locating the smallest horizontal integral projection, which was assumed to lie between 40% – 60% of the image height.

In addition to improving the accuracy of the model, this lead to elimination of user-assisted input making the process more automatic. Fourth-degree polynomial curve fitting was applied to a set of obtained horizontal integral projections to produce a smooth curve representing the average location of the gap valley.

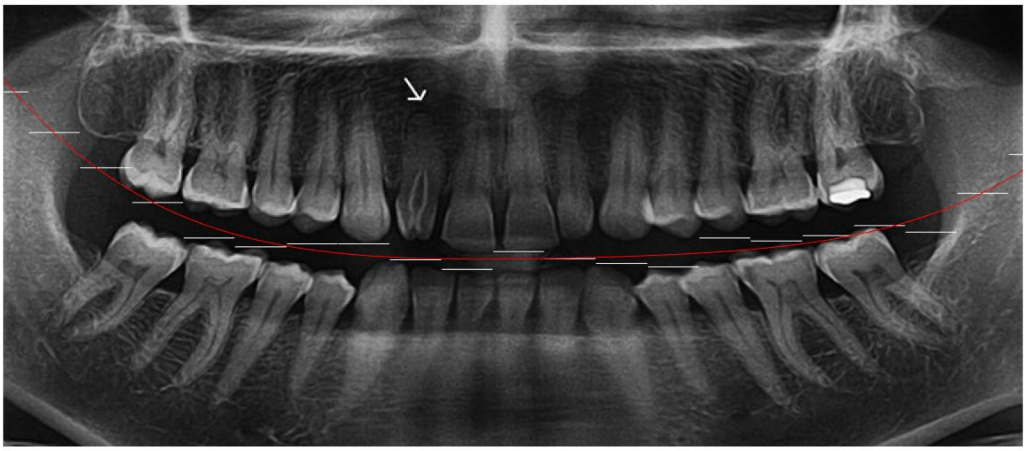


Figure 13: Smooth gap valley (red) obtained by applying polynomial curve fitting to horizontal integral projections (white)

4.2.3 Localizing Areas between Necks of Teeth

This section is going to describe the process of determining teeth necks position. Neck is a part of the tooth connecting the roots and crowns (PubMed Health, 2017). As crowns are mostly larger than other teeth parts, they often tend to occlude with each other making it difficult to separate roots from bones and soft tissues. Nevertheless, the areas between teeth necks are distinct enough and thus, could be identified on the radiograph. In addition, dental pulp (inner part of the crown) often has lower brightness compared to the enamel. As the necks are located on the same height as dental pulp, it is possible to find a line passing through both tooth elements. Such line would have the lowest integrated intensity due to low brightness of the pulp and areas between necks. The simplest way to obtain this line is to translate the gap valley vertically and sum up the intensities along this line. The smallest value would indicate the line that passes through dental pulp and dark areas between necks.

In order to find a precise location of the desired line it is important to define a translation range to reject the lines that are too close and too far from the gap valley. Applying the limits to this step would result into two values, positive and negative, indicating the number of pixels by which the line separating upper and lower jaws should be moved in order to obtain the curves passing through dental pulp of maxillary and mandibular teeth.

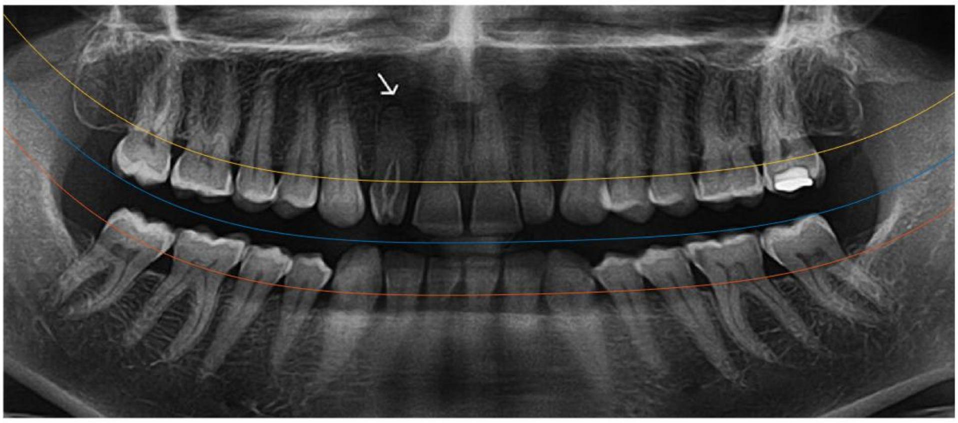


Figure 14: Representation of gap valley (blue) and lines passing through upper (yellow) and lower (amber) dental pulps

The next step of segmentation process is choosing the points representing the gaps between necks of the adjacent teeth. Since these regions have similar darkness to the gap valley, they could be found using intensity plot as well. The image was inverted, using "imcomplement" function, to show low intensity areas as peaks rather than valleys. To remove false peaks and locate single maximum point, Gaussian smoothening filter was applied. The resulting intensity profile along two curves was analysed (Figure 14). It could be observed that the points of lowest intensity (presented as peaks on the graph) correspond to the gaps between teeth (shown as red squares). Due to large intensity variation between teeth and neck areas, it could be established that the determined positions of gaps provide good approximation of separation points between teeth. Therefore, their respective coordinates were computed.

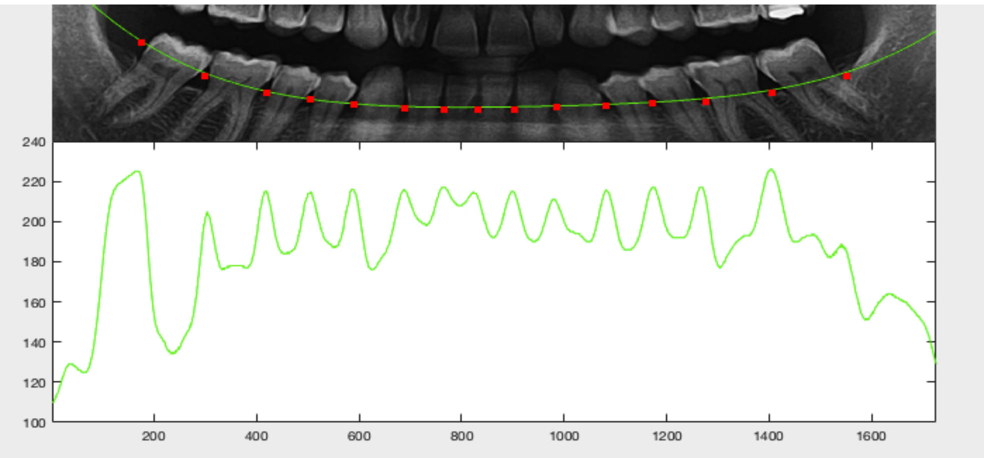


Figure 15: Located gaps between teeth necks of lower jaw (top) using intensity profile of inverted image (bottom).

4.2.4 Tooth Isolation

Although, the gaps between necks were determined, a line separating the teeth could not be drawn because the its direction was unknown. Thus, second point was obtained by moving the GV for the distance of a quarter of an average crown height. The produced line allowed to compute the gradient of the line connecting these two point. Furthermore, in order to connect this stripe with the gap valley an average width of each tooth was calculated based on the data sample used for ROI definition. As teeth are usually positioned fairly symmetrically on both sides of the jaws, only 8 values were determined. However, this assumption will not be valid for complex teeth irregularities, which would be a limitation of this design. Nevertheless, in order to reduce the likelihood of this happening, the function was set to allow tolerance of 50% in both left and right side. Hence, having defined the point between necks, gradient of the line separating adjacent teeth and approximate intersection of this line with the gap valley, GV could be connected to the top/bottom end of the image via separation line. In effect, the region bounded by two neighbouring lines and the GV constituted into a small region of interest with one tooth at maximum. As could be seen on Figure 14, panoramic radiograph was segmented into individual ROI's containing exactly one tooth, as required by the objectives.

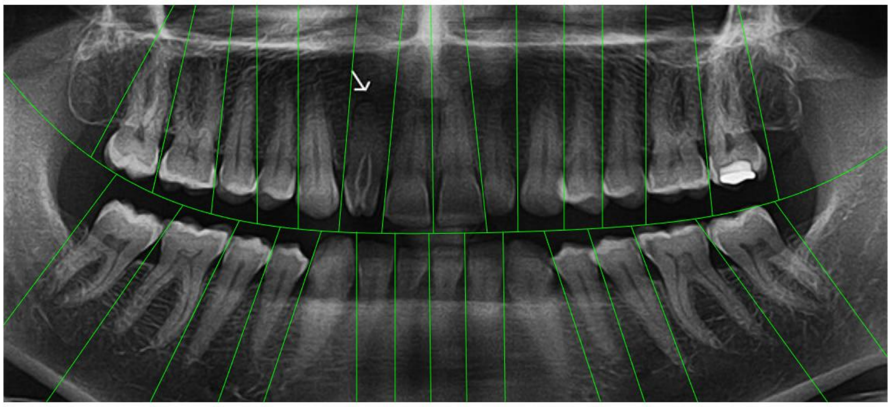


Figure 16: The resulting image after all teeth were isolated using Intelligent Segmentation

4.2.5 Measurement of Teeth Angles

Having defined the location of each tooth on the image, their corresponding angles with respect to horizontal axis could be calculated using two adjacent lines of separation. Since they pass exactly through the space between necks and do not cross with other teeth, the average value of their inclinations represents a good approximation of the angular position of corresponding tooth.

5 Evaluation and Future Work

5.1 Results

After completing the software development stage, the patient's data found online was tested in order to calculate angular position of teeth.

First, each tooth was assigned a number according to Figure 17. The patient, whose radiograph was taken for examination, did not have wisdom teeth (shown as black). Thus, 28 teeth were expected to be identified by the program.

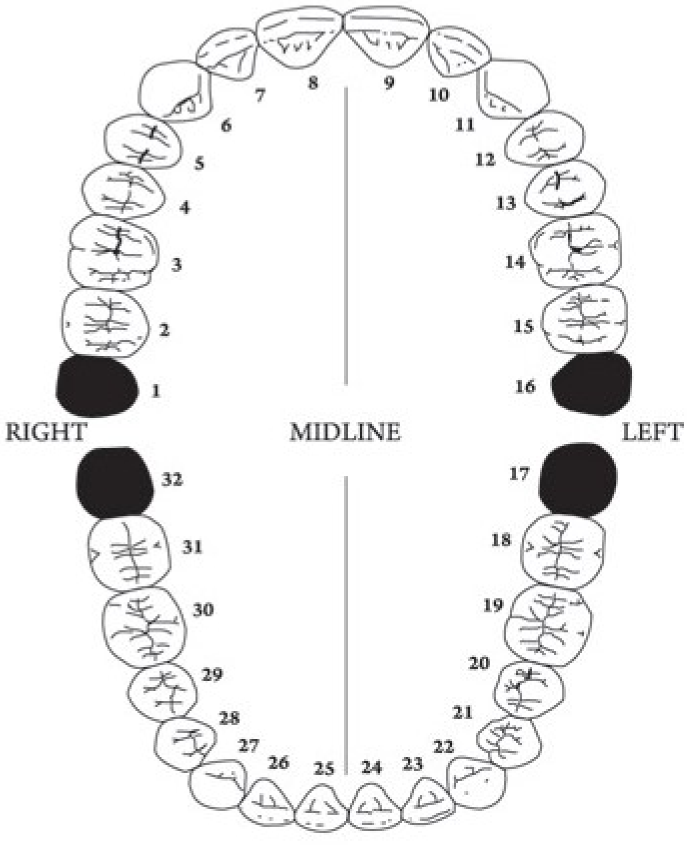


Figure 17: The schematic of tooth numbering system used for teeth identification (The Lanap and Implant Center of Pensilvania, 2017).

The image was fed into developed software and algorithm produced following measurements available in Table 1. The presented results show the tooth number and corresponding angle in degrees measured with respect to horizontal axis.

Upper Jaw		**Lower Jaw**	
Tooth №	**Angle (°)**	**Tooth №**	**Angle (°)**
2	72.255	18	127.569
3	82.147	19	109.654
4	84.289	20	108.435
5	84.489	21	109.026
6	95.711	22	109.654
7	84.289	23	90.000
8	84.189	24	90.000
9	95.711	25	57.518
10	99.807	26	57.521
11	84.289	27	35.550
12	85.289	28	70.974
13	93.814	29	67.380
14	84.289	30	59.349
15	95.711	31	52.696

Table 1: Angular position of teeth measured using Intelligent Segmentation algorithm

5.2 Critical Analysis

In order to validate the results produced by the developed algorithm, manual measurements of teeth angles were done using PixelStick software. The obtained data could be found in Appendix E. The analysis showed that the maximum error was 7.6%, while the average errors were 1.89% and 3.93% for upper and lower jaws respectively, which shows that the algorithm was fairly accurate and produced satisfactory results. Furthermore, the percentage difference between digital and manual measurements for each tooth in the upper and lower jaws was plotted (refer to Graph 1 and Graph 2). It could be observed that the error increases significantly from tooth 4 to 5 and drops from 6 to 8 at almost the same rate. Similar behaviour could be observed from teeth 8-14. Moreover, analysis of lower jaw measurements demonstrates that the error changes with respect to the tooth number analogously. As tooth number indicates its position in the jaw (Figure 17), it could be established that the error is the lowest at molars and grows between premolars and canines returning back to its lowest at incisors. This relationship could be explained by analysing the results present on Figure 16. If the teeth have wide gaps between them, the function is capable of finding precise line of separation, which could be used for angle measurements. However, when the position of

teeth in the arch deviates from simple line (usually at canines) the inclination of two lines of separation varies significantly (tooth 5 and 9) making the output less accurate.
Nevertheless, it should be noted that manual measurements were conducted without software assisting. In addition, when measuring angles, the centre of the tooth was taken as a reference, while the use of the line separating tooth into equal areas seems to be more accurate. Furthermore, human error cannot be disregarded meaning that the true error might slightly be different. Moreover, the output of the function was considerably faster (7 sec) than projected runtime (5 min). Hence, initial objective was completely satisfied.

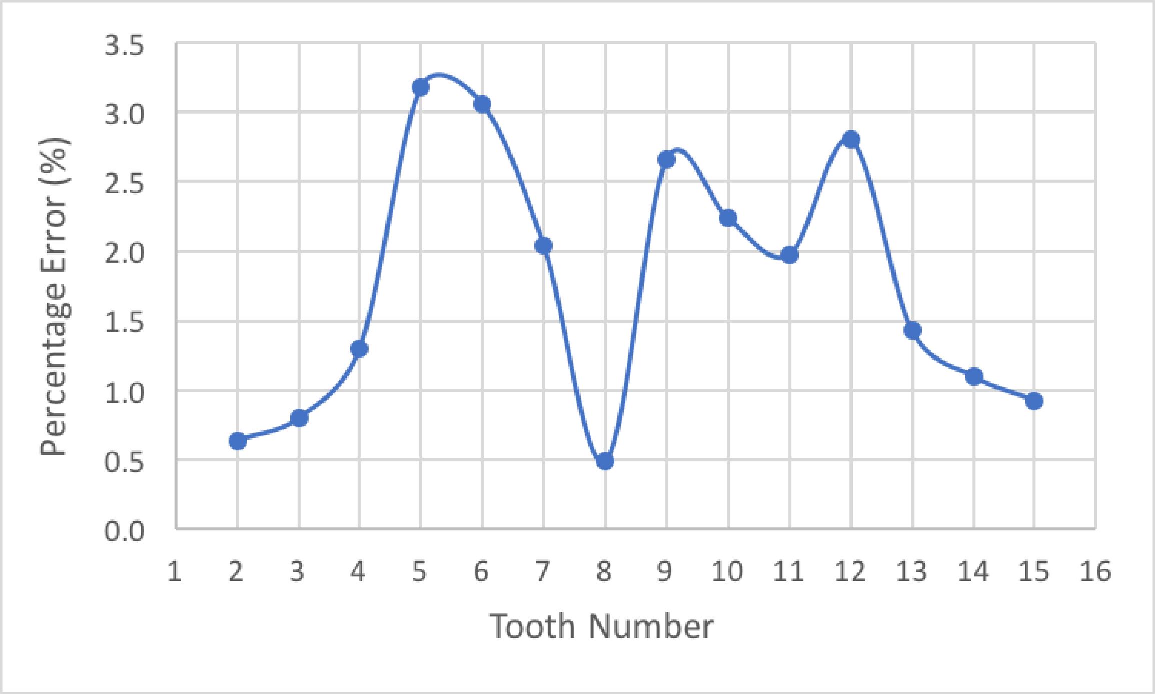


Figure 18: Percentage error between automatic and manual measurements of tooth inclinations in the upper jaw for corresponding tooth number

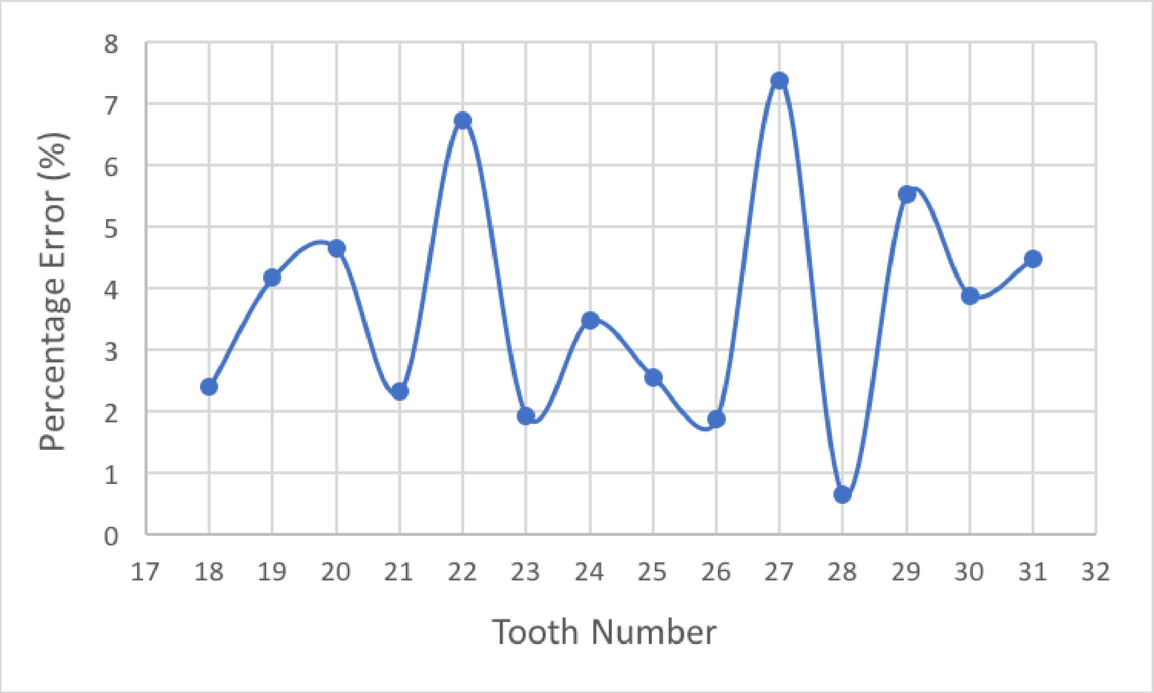


Figure 19: Percentage error between automatic and manual measurements of tooth inclinations in the lower jaw for corresponding tooth number

5.3 Future Work

Despite not producing acceptable results, the first method seems promising and could be improved. Firstly, this technique requires large database to recognize teeth. Thus, the size should be increased significantly to account for most teeth variations on various radiographs. Secondly, the teeth could be separated using second method prior finding edges in order to simplify the process. Lastly, individual boundaries could be obtained using active contours algorithm, which could then be used for calculation of angles and dimensions.

Although, the intelligent segmentation method produced satisfactory results, there are a few next steps that can be carried forward to improve the proposed algorithm. For accurate detection of separating line between teeth, especially near canines, the second point could be obtained by a more rigorous approach. Finding integral projections along a line equal to the height of average tooth would probably provide better results. In addition, scanning projections at different angles would improve precision as well. This algorithm could also be used as a first step for further image analysis. For example: teeth boundaries could be extracted using edge detection method, described in this report, to improve the results and also calculate the dimensions of teeth. Although, the quality of produced results was validated using experimental method, pattern recognition should be introduced to further improve the accuracy of this method.

6 Conclusion

In conclusion, this work was accomplished for dental orthodontic diagnosis. It has possibility to drastically improve the efficiency of existing procedure of orthodontic treatment. This paper presented two methods to perform automatic measurements of the teeth angles on panoramic radiographs.

The objectives set out at the beginning of the project were fully achieved using developed Intelligent Segmentation method (full code could be found in Appendix F). The algorithm allowed to automatically measure angles of all teeth present on the image within 7 seconds. It uses a novel approach to teeth identification problem, which involves calculation of the line separating upper and lower jaws (gap valley). The image was cut into 21 equally-sized strips, where each strip is considered as a number of horizontal lines of unit width. Intensity along each line was summed up to find integral projection. The smallest number corresponded to the lowest overall brightness, thus, was assumed to lie within the gap valley. Then, polynomial curve fitting was used to combine all horizontal lines into an approximate gap valley curve. It was then used to find the gaps between necks of teeth by translating the curve iteratively towards upper and lower jaws and plotting intensity distribution along these lines. Having located these gaps, the teeth were separated individually and their corresponding angles were determined by calculating average inclination of two adjacent separating lines. The method produced meaningful results, when was tested with patient's data. They were compared with manual measurements of teeth angles obtained using computer software. The determined error of 7.9% was small enough to prove the accuracy of proposed method.

Another approach, that was tested in this paper, did not produce satisfactory results. The following approach was taken: 1) The image was pre-processed to remove noise (usually present on radiographs) and improve overall contrast of images. 2) Enhanced image was used to reveal teeth boundaries. 3) These edges were then fed into Artificial Neural Networks system, which compared them with database created to teach it recognize tooth shape. The shape matching process was accomplished by setting up firing rules that allowed to compare large set of data for presence of similarities. Failure to produce acceptable results could be explained by the overall complexity of analysed data and distortion caused by surrounding noise. In addition, the size of the training database was not large enough to account for all possible variations in shape and orientation of teeth. Despite failing to identify teeth on the image, several key experiences were obtained. Firstly, different pre-processing techniques were studied (noise removal, edge detection and others), which allowed to strengthen understanding of computer vision field. In addition, the method utilized pattern recognition and machine learning techniques, which are considered the future of medical imaging and could be implemented in conjunction with the first method. Thus, this method is considered promising and further research in the field of machine learning would help to improve it.

7 References

• Abanto, J., Carvalho, T., Mendes, F., Wanderley, M., Bönecker, M. and Raggio, D. (2010). Impact of oral diseases and disorders on oral health-related quality of life of preschool children. Community Dentistry and Oral Epidemiology, 39(2), pp.105-114.

• Budinger, T. and Brahme, A. (2014). Comprehensive biomedical physics. 1st ed. Oxford: Elsevier, pp.234-235.

• Chen, J., Benesty, J., Huang, Y. and Doclo, S. (2006). New insights into the noise reduction Wiener filter. IEEE Transactions on Audio, Speech and Language Processing, 14(4), pp.1218-1234.

• Cordray, F.E. (2016) 'Articulated dental cast analysis of asymptomatic and symptomatic populations', International Journal of Oral Science, 8(2), pp. 126–132. doi: 10.1038/ijos.2015.44.

• Demuth, H. and Beale, M. (no date) Neural Network Toolbox. Available at: `http://www.image.ece.ntua.gr/courses_static/nn/matlab/nnet.pdf` [Accessed 6 Dec. 2016]

• DrDerhamdental.com. (2017). Panoramic Radiograph. [online] Available at: `http://drderhamdental.com/services/dental-hygiene-and-prevention/panoramic-radiographs/` [Accessed 10 Mar. 2017].

• El Joumani, S., Mechkouri, S., Zennouhi, R., El Kadmiri, O. and Masmoudi, L. (2017). Segmentation method based on multiobjective optimization for very high spatial resolution satellite images. EURASIP Journal on Image and Video Processing, 2017(1).

• Gao, W., Zhang, X., Yang, L. and Liu, H. (2010). An improved Sobel edge detection. 2010 3rd International Conference on Computer Science and Information Technology, 5, pp.67-71.

• Harrell, W., Hatcher, D. and Bolt, R. (2002). In search of anatomic truth: 3-dimensional digital modeling and the future of orthodontics. American Journal of Orthodontics and Dentofacial Orthopedics, 122(3), pp.325-330.

• Hobson, R., McCabe, J. and Hogg, S. (2001). Bond strength to surface enamel for different tooth types. Dental Materials, 17(2), pp.184-189.

• Kpalma, K. and Ronsin, J. (2007). An Overview of Advances of Pattern Recognition Systems in Computer Vision. In: A. Dutta and G. Obinata, ed., An Overview of Advances of Pattern Recognition Systems in Computer Vision, 1st ed. I-Tech Education and Publishing, p.171.

• Mathworks.com. (2017). System Requirements - Release 2016a. [online] Available at:`https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/support/sysreq/files/SystemRequirements-Release2016a_Windows.pdf` [Accessed 21 Apr. 2017].

• NHS.uk. (2015). Orthodontics. [online] Available at: `http://www.nhs.uk/conditions/Orthodontics/Pages/Introduction.aspx` [Accessed 24 Apr. 2017].

• Papa, J., Falcão, A., de Albuquerque, V. and Tavares, J. (2012). Efficient supervised optimum-path forest classification for large datasets. Pattern Recognition, 45(1), pp.512-520.

- Prasad, K., Nigam, D.C., Lakhotiya, A. and Umre, D. (2013) 'Character recognition using Matlab's neural network Toolbox', International Journal Science and Technology, 6(1), pp. 13–20.
- PubMed Health. (2017). Dental Cervix (Tooth Necks). [online] Available at: `https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0025298/` [Accessed 17 Apr. 2017].
- Rad, A., Amin, I., Rahim, M. and Kolivand, H. (2015). Computer-Aided Dental Caries Detection System from X-Ray Images. Advances in Intelligent Systems and Computing, 331, pp.233-243.
- Rani, R. and Kumari, S. (2016). A Review on Edge Detection Using Different Techniques. International Journal of Science and Research (IJSR), 5(6), pp.2133-2137.
- RenewDigital.com. (2017). GENDEX ORTHORALIX 8500 DDE Panorex. [online] Available at: `http://www.renewdigital.com/gendex-orthoralix-8500-dde-panorex/` [Accessed 21 Apr. 2017].
- Rusanen, J., Silvola, A., Tolvanen, M., Pirttiniemi, P., Lahti, S. and Sipila, K. (2011). Pathways between temporomandibular disorders, occlusal characteristics, facial pain, and oral health-related quality of life among patients with severe malocclusion. The European Journal of Orthodontics, 34(4), pp.512-517.
- Rushton, V., Horner, K. and Worthington, H. (2001) 'Screening panoramic radiology of adults in general dental practice: Radiological findings', British dental journal., 190(9), pp. 495–501.
- Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, 13(1), pp.146-168.
- The Lanap and Implant Center of Pensilvania. (2017). Tooth Numbering. [online] Available at: `https://perioimplants.us/tooth-numbering/` [Accessed 24 Apr. 2017].

8 Appendices

8.1 Appendix A – System Requirements for MATLAB Release 2016a

Operating Systems	Processors	Disk Space	RAM	Graphics
Windows 10	Any Intel or AMD x86-64 processor	2 GB for MATLAB only, 4–6 GB for a typical installation	2 GB	No specific graphics card is required. Hardware accelerated, graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.
Windows 8.1				
Windows 8				
Windows 7 Service Pack 1				
Windows Server 2012				
Windows Server 2008 R2 Service Pack 1				
Windows Server 2008 Service Pack 2				

Table 2: System Requirements for Matlab Release 2016a (Mathworks.com, 2017)

8.2 Appendix B – Edge Detection Algorithm

Matlab Code of Sobel Edge Detection

```
% German Malsagov
% 24 April 2017
% german.malsagov.13@ucl.ac.uk
% function edge_detection = Sobel_edge_detection_function(img)

% Purpose:  This code allows to do edge detection on panoramic
%           radiograph using Sobel filter function
%
% Inputs:   Panoramic image
%
% Outputs:  Black&White panoramic image with detected edges of teeth

function edge_detection = Sobel_edge_detection_function(img)

% Generate a MEX function using 'codegen sobel' command

% Read the original image command
I = img;

se = strel('disk',12);
J = imsubtract(imadd(I,imtophat(I,se)),imbothat(I,se));

% This line determines the size of the original image (will be used
% later to revert to original dimensions
[x1,y1,z1] = size(J);

% This function resizes image to 1024x1024 dimensions required by
% gray function
K = imresize(I,[1024 1024]);

% Following command coverts color image into equivalent image with
% normalized values (0.0 for black, 1.0 for white)
G=(0.3*double(K(:,:,1))+0.59*double(K(:,:,1))+0.11*double(K(:,:,1)))/255;

% This command passes the normalized image through the Sobel Filter
edgeIm = sobel_mex(G, 0.18); % 0.18 gives best results
```

```
% Display the result
 Im_sobel_filter = repmat(edgeIm, [1 1 3]);

% This command converts image into grayscale
 Im_grayscale = rgb2gray(Im_sobel_filter);

% This command applies Wiener filter to reduce noise on grayscale image
 Im_Wiener_filter = wiener2(Im_grayscale,[5 5]);

% Define number of pixels of the image
 [x,y]=size(Im_Wiener_filter);
X=1:x;
Y=1:y;

% This transforms the domain specified by vectors x and y into
% arrays X and Y
 [xx,yy]=meshgrid(Y,X);

%Converts image into double precision format
 i=im2double(Im_Wiener_filter);

%This algorythm converts the filtered image to its original dimensions
KK = imresize(i,[x1,y1]);

 edge_detection = imshow(KK);

 end
```

8.3 Appendix C – Database Created for Pattern Recognition Model

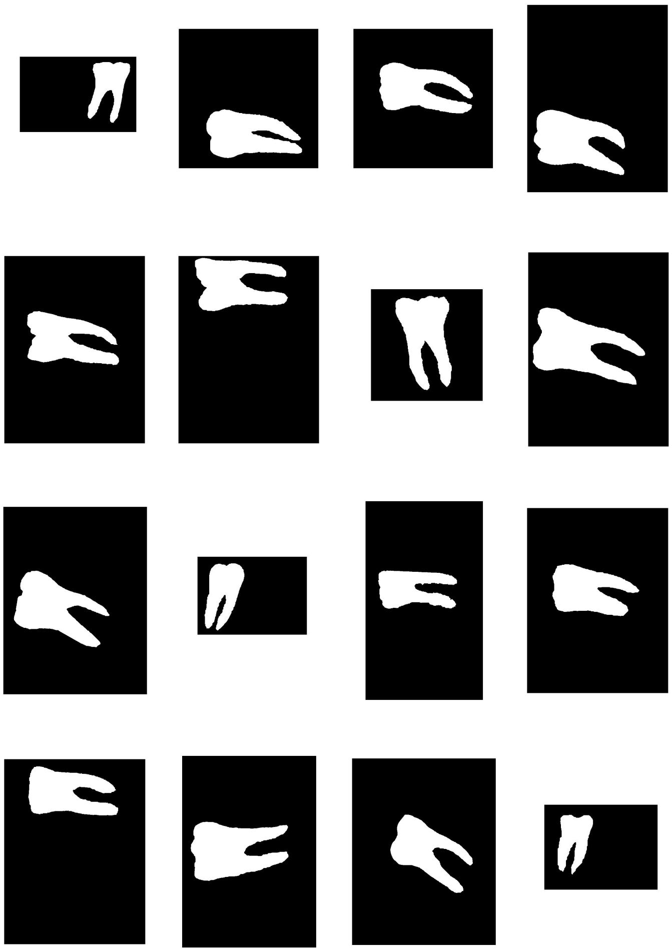


Figure 20: Database of Molar Teeth Created for Pattern Recognition Step

Appendix C – Database Created for Pattern Recognition Model (Continued)

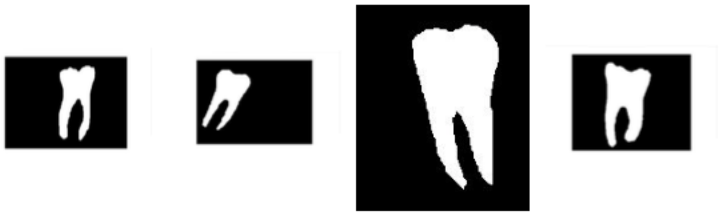


Figure 21: Database of Molar Teeth Created for Pattern Recognition Step (continued)

8.4 Appendix D – Distances Measured for Calculation of the Region of Interest

Image №	R1 [pixels]	R2 [pixels]	R3 [pixels]	R4 [pixels]
1	722	817	300	355
2	536	590	180	345
3	544	630	176	365
4	796	742	168	500
5	660	512	252	328
6	766	792	230	334
7	735	726	205	318
8	664	618	136	400
9	576	580	190	310
10	746	736	188	323
11	545	553	115	337
12	514	570	145	392
13	568	560	190	360
14	869	829	363	320
15	738	691	177	340
16	692	685	157	390
17	600	618	167	268
18	498	642	179	320
19	648	681	203	304
20	669	685	106	378
21	762	693	139	346
22	540	560	180	323

Table 3: Distance measurements for estimation of ROI size.

Appendix D – Distances Measured for Calculation of the Region of Interest (Continued)

Image №	R1 [pixels]	R2 [pixels]	R3 [pixels]	R4 [pixels]
23	716	640	286	249
24	776	591	211	343
25	733	765	300	206
26	860	806	165	375
27	837	703	317	250
28	711	700	220	320
29	608	580	175	380
30	734	713	260	250
31	756	823	213	220
32	772	654	217	356
33	572	578	170	277
34	770	666	123	328
35	566	605	190	247
36	737	709	200	340
37	710	722	223	330
38	705	723	130	285
39	628	695	185	285
40	578	570	284	280
41	707	660	250	290
42	636	753	110	378
43	840	820	165	290
44	757	757	100	357

Table 4: Distance measurements for estimation of ROI size (continued).

Appendix D – Distances Measured for Calculation of the Region of Interest (Continued)

Image №	R1 [pixels]	R2 [pixels]	R3 [pixels]	R4 [pixels]
45	530	535	150	336
46	757	817	198	230
47	636	697	185	313
48	602	608	166	332
49	648	647	120	400
50	680	707	183	320
51	850	852	330	390
52	540	560	150	387
53	767	697	313	280
54	793	755	150	445
55	530	546	90	390
56	784	680	219	300
57	606	617	85	355
58	700	720	150	330
59	640	630	220	290
60	730	760	217	350
61	800	800	255	304
62	790	790	354	294
63	560	560	170	375
64	523	523	100	349
65	538	574	151	285
66	657	713	190	325

Table 5: Distance measurements for estimation of ROI size (continued).

Appendix D – Distances Measured for Calculation of the Region of Interest (Continued)

Image №	R1 [pixels]	R2 [pixels]	R3 [pixels]	R4 [pixels]
67	685	625	211	263
68	780	750	230	359
69	737	730	227	338
70	600	619	259	285
71	654	623	170	385
72	697	723	227	325
73	867	852	275	340
74	636	666	200	323

Table 6: Distance measurements for estimation of ROI size (continued).

8.5 Appendix E – Data Measurements for Error Analysis

Upper Jaw			
	Digital	**Physical**	
Tooth №	**Angle (°)**	**Angle (°)**	**Error (%)**
2	72.255	72.719	0.638
3	82.147	81.496	0.798
4	84.289	84.401	0.132
5	84.289	82.661	1.970
6	95.711	91.871	4.179
7	84.289	82.606	2.038
8	84.189	83.782	0.486
9	95.711	94.234	1.567
10	89.637	86.676	3.416
11	84.289	86.987	3.101
12	85.289	87.745	2.799
13	92.814	91.507	1.428
14	92.367	91.364	1.098
15	99.504	98.592	0.925

Table 7: Comparison of angular results (upper jaw) produced by Intelligent Segmentation and manual measurements using software

Appendix E – Data Measurements for Error Analysis (Continued)

Lower Jaw			
	Digital	**Physical**	
Tooth №	**Angle (°)**	**Angle (°)**	**Error (%)**
18	130.684	127.569	2.384
19	114.416	109.654	4.162
20	118.635	110.276	7.046
21	112.628	109.026	3.198
22	106.566	109.654	2.898
23	88.299	90.000	1.926
24	86.986	90.000	3.465
25	83.386	85.518	2.557
26	85.914	87.521	1.871
27	72.747	78.111	7.373
28	71.436	70.974	0.646
29	63.858	67.380	5.516
30	61.745	59.349	3.880
31	50.44	52.696	4.473

Table 8: Comparison of angular results (lower jaw) produced by Intelligent Segmentation and manual measurements using software

8.6 Appendix F – Matlab Code for Intelligent Segmentation Algorithm

Matlab Code

```
% German Malsagov
% 24 April 2017
% german.malsagov.13@ucl.ac.uk
% function angular_data = intelligent_segmentation(panoramic_image)

% Purpose:  This function calculates predicted angular information
%           about patient's teeth using following sequence:
%           - Computing line between upper and lower jaws
%           - Gaps betweens teeth necks are located
%           - Individual teeth are separated using gap locations
%           - Teeth angles are calculated using two adjacent lines
%
% Inputs:   Panoramic image
%           Dimensions of Region of Interest (ROI)
%
% Outputs: Angular positions of each tooth in degrees

 function angular_data = intelligent_segmentation(panoramic_image)

% Dimensions of Region of Interest
R1 = 875; %pixels
R2 = 850; %pixels
R3 = 318; %pixels
R4 = 428; %pixels

% Defining Input Image
I = imread(panoramic_image);

% Resizing image to fit standard ROI
 fig = imresize(I,[1200 2400]);
 [y, x, z] = size(fig);

%% Top_Bottom Hat filtering (if image has poor contrast)
%{
 se = strel('disk',12);
 tophat = imtophat(fig,se);
bothat = imbothat(fig,se);
K = imsubtract(imadd(fig,imtophat(fig,se)),imbothat(fig,se));
%}
%% Splitting image into 21 vertical strips

% Cropping Image to obtain ROI according to R1,R2,R3,R4
J=fig(0.5*y-R3:0.5*y+R4, 0.5*x-R2:0.5*x+R1);

% Pre-defining variables to avoid repeated calculations

% Measuring size of analysed image
[y1, x1] = size(J);

% Computing the centre of image
x_h = 0.5*x1;
y_h = 0.5*y1;

% Computing cetral stripe (round function is used to avoid error)
I1 = J(1:y1, x_h-round(1/40*x1):x_h+round(1/40*x1));

% Right Hand Side Stipes
I2 = J(1:y1, x_h+round(1/40*x1)+1:x_h+round(3/40*x1));
I3 = J(1:y1, x_h+round(3/40*x1)+1:x_h+round(5/40*x1));
I4 = J(1:y1, x_h+round(5/40*x1)+1:x_h+round(7/40*x1));
I5 = J(1:y1, x_h+round(7/40*x1)+1:x_h+round(9/40*x1));
I6 = J(1:y1, x_h+round(9/40*x1)+1:x_h+round(11/40*x1));
I7 = J(1:y1, x_h+round(11/40*x1)+1:x_h+round(13/40*x1));
I8 = J(1:y1, x_h+round(13/40*x1)+1:x_h+round(15/40*x1));
I9 = J(1:y1, x_h+round(15/40*x1)+1:x_h+round(17/40*x1));
I10 = J(1:y1, x_h+round(17/40*x1)+1:x_h+round(19/40*x1));
I11 = J(1:y1, x_h+round(19/40*x1)+1:x1);

% Left Hand Side Stripes
I12 = J(1:y1, x_h-round(3/40*x1):x_h-round(1/40*x1)-1);
I13 = J(1:y1, x_h-round(5/40*x1):x_h-round(3/40*x1)-1);
I14 = J(1:y1, x_h-round(7/40*x1):x_h-round(5/40*x1)-1);
I15 = J(1:y1, x_h-round(9/40*x1):x_h-round(7/40*x1)-1);
I16 = J(1:y1, x_h-round(11/40*x1):x_h-round(9/40*x1)-1);
I17 = J(1:y1, x_h-round(13/40*x1):x_h-round(11/40*x1)-1);
```

```
I18 = J(1:y1, x_h-round(15/40*x1):x_h-round(13/40*x1)-1);
I19 = J(1:y1, x_h-round(17/40*x1):x_h-round(15/40*x1)-1);
I20 = J(1:y1, x_h-round(19/40*x1):x_h-round(17/40*x1)-1);
I21 = J(1:y1, 1:x_h-round(19/40*x1)-1);
%}

%% Displaying all strips on one figure
%{
figure
subplot(1, 21, 1,'align'), imshow(I21);
subplot(1, 21, 2,'align'), imshow(I20);
subplot(1, 21, 3,'align'), imshow(I19);
subplot(1, 21, 4,'align'), imshow(I18);
subplot(1, 21, 5,'align'), imshow(I17);
subplot(1, 21, 6,'align'), imshow(I16);
subplot(1, 21, 7,'align'), imshow(I15);
subplot(1, 21, 8,'align'), imshow(I14);
subplot(1, 21, 9,'align'), imshow(I13);
subplot(1, 21, 10,'align'), imshow(I12);
subplot(1, 21, 11,'align'), imshow(I1);
subplot(1, 21, 12,'align'), imshow(I2);
subplot(1, 21, 13,'align'), imshow(I3);
subplot(1, 21, 14,'align'), imshow(I4);
subplot(1, 21, 15,'align'), imshow(I5);
subplot(1, 21, 16,'align'), imshow(I6);
subplot(1, 21, 17,'align'), imshow(I7);
subplot(1, 21, 18,'align'), imshow(I8);
subplot(1, 21, 19,'align'), imshow(I9);
subplot(1, 21, 20,'align'), imshow(I10);
%subplot(1, 21, 21,'align'), imshow(I11);
align_Ylabels(gcf)
%}


%% Finding Horizontal Projections
% Output of the functions local_min/2/edge represents 2
% vectors containing coordinates of horizontal projections
% local_min function is applied to the mid strip. It is
% looking for the jaw centre which usually lies between
% 40%-60% height
%
% Mid strip
output1 = local_minn(I1);
xline1 = output1(1,:);
yline1 = output1(2,:);

% Right Hand Side
output2 = local_min22(I2);
xline2 = output2(1,:);
yline2 = output2(2,:);

output3 = local_min22(I3);
xline3 = output3(1,:);
yline3 = output3(2,:);

output4 = local_min22(I4);
xline4 = output4(1,:);
yline4 = output4(2,:);

output5 = local_min22(I5);
xline5 = output5(1,:);
yline5 = output5(2,:);

output6 = local_min22(I6);
xline6 = output6(1,:);
yline6 = output6(2,:);

output7 = local_min22(I7);
xline7 = output7(1,:);
yline7 = output7(2,:);

output8 = local_min22(I8);
xline8 = output8(1,:);
yline8 = output8(2,:);

output9 = local_min22(I9);
xline9 = output9(1,:);
yline9 = output9(2,:);
```

```
output10 = local_min_edge2(I10,yline9);
xline10 = output10(1,:);
yline10 = output10(2,:);

output11 = local_min_edge2(I11,yline10);
xline11 = output11(1,:);
yline11 = output11(2,:);

% Left Hand Side
output12 = local_min22(I12);
xline12 = output12(1,:);
yline12 = output12(2,:);

output13 = local_min22(I13);
xline13 = output13(1,:);
yline13 = output13(2,:);

output14 = local_min22(I14);
xline14 = output14(1,:);
yline14 = output14(2,:);

output15 = local_min22(I15);
xline15 = output15(1,:);
yline15 = output15(2,:);

output16 = local_min22(I16);
xline16 = output16(1,:);
yline16 = output16(2,:);

output17 = local_min22(I17);
xline17 = output17(1,:);
yline17 = output17(2,:);

output18 = local_min22(I18);
xline18 = output18(1,:);
yline18 = output18(2,:);

output19 = local_min_edge2(I19,yline18);
```

```
xline19 = output19(1,:);
yline19 = output19(2,:);

output20 = local_min_edge2(I20,yline19);
xline20 = output20(1,:);
yline20 = output20(2,:);

output21 = local_min_edge2(I21,yline20);
xline21 = output21(1,:);
yline21 = output21(2,:);

%}


%% Applying Polynomial Curve Fitting

%
% Defining x-coordinates of Gap Valley by adding last term
% of previous IP to the next one
x_I21 = xline21;
x_I20 = x_I21(end)+xline20;
x_I19 = x_I20(end)+xline19;
x_I18 = x_I19(end)+xline18;
x_I17 = x_I18(end)+xline17;
x_I16 = x_I17(end)+xline16;
x_I15 = x_I16(end)+xline15;
x_I14 = x_I15(end)+xline14;
x_I13 = x_I14(end)+xline13;
x_I12 = x_I13(end)+xline12;
x_I1 = x_I12(end)+xline1;
x_I2 = x_I1(end)+xline2;
x_I3 = x_I2(end)+xline3;
x_I4 = x_I3(end)+xline4;
x_I5 = x_I4(end)+xline5;
x_I6 = x_I5(end)+xline6;
x_I7 = x_I6(end)+xline7;
x_I8 = x_I7(end)+xline8;
x_I9 = x_I8(end)+xline9;
x_I10 = x_I9(end)+xline10;
```

```
x_I11 = x_I10(end)+xline11;
%}
% Defining vector a which represents x-coordinates of gap valley
a1 = [x_I21 x_I20 x_I19 x_I18 x_I17 x_I16 x_I15 x_I14 x_I13 x_I12];
a2 = [x_I1 x_I2 x_I3 x_I4 x_I5 x_I6 x_I7 x_I8 x_I9 x_I10 x_I11];
a = [a1 a2];

% Defining vector b which represents y-coordinates of gap valley
b1 = [yline21 yline20 yline19 yline18 yline17 yline16 yline15];
b2 =  [yline14 yline13 yline12 yline1 yline2 yline3 yline4];
b3 =  [yline5 yline6 yline7 yline8 yline9 yline10 yline11];
b = [b1 b2 b3];
% Applying polifit function with 4th degree polynomial
[p,~,mu] = polyfit(a,b,4);

% Plotting Gap Valley using polyval
f = polyval(p,a,[],mu);

% Rounding values of f to find intensity values for further
% analysis of gaps
ff = round(f);

% Plotting Gap Valleys
%{
hold on
imshow(J);
hold on
plot(a,f)
hold on
plot(a,f+95);
hold on
plot(a,f-110);
%}

%}
%% Plotting Horizontal Projections
%{
imshow(J);
hold on
```

```
line(x_I21,yline21,'color', 'white');
hold on
line(x_I20,yline20,'color', 'white');
hold on
line(x_I19,yline19,'color', 'white');
hold on
line(x_I18,yline18,'color', 'white');
hold on
line(x_I17,yline17,'color', 'white');
hold on
line(x_I16,yline16,'color', 'white');
hold on
line(x_I15,yline15,'color', 'white');
hold on
line(x_I14,yline14,'color', 'white');
hold on
line(x_I13,yline13,'color', 'white');
hold on
line(x_I12,yline12,'color', 'white');
hold on
line(x_I1,yline1,'color', 'white');
hold on
line(x_I2,yline2,'color', 'white');
hold on
line(x_I3,yline3,'color', 'white');
hold on
line(x_I4,yline4,'color', 'white');
hold on
line(x_I5,yline5,'color', 'white');
hold on
line(x_I6,yline6,'color', 'white');
hold on
line(x_I7,yline7,'color', 'white');
hold on
line(x_I8,yline8,'color', 'white');
hold on
line(x_I9,yline9,'color', 'white');
hold on
line(x_I10,yline10,'color', 'white');
```

```
hold on
line(x_I11,yline11,'color', 'white');
hold on
plot(a,f,'color','red');


%}

%% Finding Gaps between Teeth Necks

%{
% Image filtering
IM = imcomplement(J);
NHOOD=[1, 0, 0; 0, 1, 0; 0, 0, 1];
IM2 = rangefilt(IM,NHOOD);
IM3 = immultiply(IM,IM2);
imshow(IM3);
%}
%
% Pre-allocating memory
int_table = zeros(length(a),4);
tble = zeros(length(a),4);

%% Computing Location of Gaps between Necks of Lower Jaw Teeth
% Defining 1st set of coordinates along which intensity is
% calculated
ff1 = ff+80;
for i = 1:length(a)
    x = a(i);
    y = ff1(i);
    int_table(i,1) = J(y,x);
end

% Defining 2nd set of coordinates along which intensity is
% calculated
ff2 = ff+95;
for j = 1:length(a)
    x = a(j);
    y = ff2(j);
    int_table(j,2) = J(y,x);
end

% Defining vector with indices of points corresponding to lowest intensity
ind_80=[185;307;422;510;590;690;765;835;905;980;1080;1170;1265;1400;1535];
ind_95=[177;299;415;505;585;690;765;835;905;980;1085;1175;1270;1410;1545];

%% Segmenting Individual Teeth of Lower Jaw
%
% Defining coordinates of points with lowest intensity [e(index) -
% horizontal, r(index) - vertical]
e1 = [a(ind_80(1)),a(ind_95(1))];
r1 = [ff1(ind_80(1)),ff2(ind_95(1))];
slope1 = (r1(1) - r1(2))/(e1(1) - e1(2));
x_31 = a(ind_95(1))-(ff2(ind_95(1))-y1)/slope1;
e_12 = [a(1+220), x_31];
r_12 = [ff(1+220), y1];

e2 = [a(ind_80(2)),a(ind_95(2))];
r2 = [ff1(ind_80(2)),ff2(ind_95(2))];
slope2 = (r2(1) - r2(2))/(e2(1) - e2(2));
x_32 = a(ind_95(2))-(ff2(ind_95(2))-y1)/slope2;
e22 = [a(1+350), x_32];
r22 = [ff(1+350), y1];

e3 = [a(ind_80(3)),a(ind_95(3))];
r3 = [ff1(ind_80(3)),ff2(ind_95(3))];
slope3 = (r3(1) - r3(2))/(e3(1) - e3(2));
x_33 = a(ind_95(3))-(ff2(ind_95(3))-y1)/slope3;
e32 = [a(1+460), x_33];
r32 = [ff(1+460), y1];

e4 = [a(ind_80(4)),a(ind_95(4))];
r4 = [ff1(ind_80(4)),ff2(ind_95(4))];
slope4 = (r4(1) - r4(2))/(e4(1) - e4(2));
x_34 = a(ind_95(4))-(ff2(ind_95(4))-y1)/slope4;
e42 = [a(1+540), x_34];
r42 = [ff(1+540), y1];
```

```
e5 = [a(ind_80(5)),a(ind_95(5))];
r5 = [ff1(ind_80(5)),ff2(ind_95(5))];
slope5 = (r5(1) - r5(2))/(e5(1) - e5(2));
x_35 = a(ind_95(5))-(ff2(ind_95(5))-y1)/slope5;
e52 = [a(1+620), x_35];
r52 = [ff(1+620), y1];

e6 = [a(ind_80(6)),a(ind_95(6))];
r6 = [ff1(ind_80(6)),ff2(ind_95(6))];
slope6 = (r6(1) - r6(2))/(e6(1) - e6(2));
x_36 = a(ind_95(6))-(ff2(ind_95(6))-y1)/slope6;
e62 = [a(ind_80(6)), x_36];
r62 = [ff(x1-1080), y1];

e7 = [a(ind_80(7)),a(ind_95(7))];
r7 = [ff1(ind_80(7)),ff2(ind_95(7))];
slope7 = (r7(1) - r7(2))/(e7(1) - e7(2));
x_37 = a(ind_95(7))-(ff2(ind_95(7))-y1)/slope7;
e72 = [a(ind_80(7)), x_37];
r72 = [ff(x1-990), y1];

e8 = [a(ind_80(8)),a(ind_95(8))];
r8 = [ff1(ind_80(8)),ff2(ind_95(8))];
slope8 = (r8(1) - r8(2))/(e8(1) - e8(2));
x_38 = a(ind_95(8))-(ff2(ind_95(8))-y1)/slope8;
e82 = [a(x1-895), x_38];
r82 = [ff(x1-895), y1];

e9 = [a(ind_80(9)),a(ind_95(9))];
r9 = [ff1(ind_80(9)),ff2(ind_95(9))];
slope9 = (r9(1) - r9(2))/(e9(1) - e9(2));
x_39 = a(ind_95(9))-(ff2(ind_95(9))-y1)/slope9;
e92 = [a(x1-825), x_39];
r92 = [ff(x1-825), y1];

e10 = [a(ind_80(10)),a(ind_95(10))];
r10 = [ff1(ind_80(10)),ff2(ind_95(10))];
slope10 = (r10(1) - r10(2))/(e10(1) - e10(2));
x_310 = a(ind_95(10))-(ff2(ind_95(10))-y1)/slope10;
```

```
e102 = [a(x1-750), x_310];
r102 = [ff(x1-750), y1];

e11 = [a(ind_80(11)),a(ind_95(11))];
r11 = [ff1(ind_80(11)),ff2(ind_95(11))];
slope11 = (r11(1) - r11(2))/(e11(1) - e11(2));
x_311 = a(ind_95(11))-(ff2(ind_95(11))-y1)/slope11;
e112 = [a(x1-670), x_311];
r112 = [ff(x1-670), y1];

e12 = [a(ind_80(12)),a(ind_95(12))];
r12 = [ff1(ind_80(12)),ff2(ind_95(12))];
slope12 = (r12(1) - r12(2))/(e12(1) - e12(2));
x_312 = a(ind_95(12))-(ff2(ind_95(12))-y1)/slope12;
e122 = [a(x1-585), x_312];
r122 = [ff(x1-585), y1];

e13 = [a(ind_80(13)),a(ind_95(13))];
r13 = [ff1(ind_80(13)),ff2(ind_95(13))];
slope13 = (r13(1) - r13(2))/(e13(1) - e13(2));
x_313 = a(ind_95(13))-(ff2(ind_95(13))-y1)/slope13;
e132 = [a(x1-500), x_313];
r132 = [ff(x1-500), y1];

e14 = [a(ind_80(14)),a(ind_95(14))];
r14 = [ff1(ind_80(14)),ff2(ind_95(14))];
slope14 = (r14(1) - r14(2))/(e14(1) - e14(2));
x_314 = a(ind_95(14))-(ff2(ind_95(14))-y1)/slope14;
e142 = [a(x1-375), x_314];
r142 = [ff(x1-375), y1];

e15 = [a(ind_80(15)),a(ind_95(15))];
r15 = [ff1(ind_80(15)),ff2(ind_95(15))];
slope15 = (r15(1) - r15(2))/(e15(1) - e15(2));
x_315 = a(ind_95(15))-(ff2(ind_95(15))-y1)/slope15;
e152 = [a(x1-220), x_315];
r152 = [ff(x1-220), y1];
```

```
%% Computing Location of Gaps between Necks (Lower Upper Teeth)

% Defining 1st set of coordinates along which intensity
% is calculated
ff3 = ff-100;
for j = 1:length(a)
    x = a(j);
    y = ff3(j);
    int_table(j,3) = J(y,x);
end

% Defining 1st set of coordinates along which intensity
% is calculated
ff4 = ff-110;
for j = 1:length(a)
    x = a(j);
    y = ff4(j);
    int_table(j,4) = J(y,x);
end

% Defining locations of gaps between teeth necks
ind_100=[212;317;420;494;577;659;748;835;930;1001;1094;1169;1254;1374;1490];
ind_110=[216;319;421;495;576;660;749;834;929;1002;1095;1170;1255;1373;1488];

%% Segmenting Individual Teeth of Upper Jaw
% Defining coordinates of points with lowest intensity [e(index) -
% horizontal, r(index) - vertical]
e1_u = [a(ind_100(1)),a(ind_110(1))];
r1_u = [ff3(ind_100(1)),ff4(ind_110(1))];
slope1_u = (r1_u(1) - r1_u(2))/(e1_u(1) - e1_u(2));
x_31_u = a(ind_110(1))-(ff4(ind_110(1))-1)/slope1_u;
e_12_u = [a(1+170), x_31_u];
r_12_u = [ff(1+170), 1];

e2_u = [a(ind_100(2)),a(ind_110(2))];
r2_u = [ff3(ind_100(2)),ff4(ind_110(2))];
slope2_u = (r2_u(1) - r2_u(2))/(e2_u(1) - e2_u(2));
x_32_u = a(ind_110(2))-(ff4(ind_110(2))-1)/slope2_u;
e22_u = [a(1+290), x_32_u];
```

```
r22_u = [ff(1+290), 1];

e3_u = [a(ind_100(3)),a(ind_110(3))];
r3_u = [ff3(ind_100(3)),ff4(ind_110(3))];
slope3_u = (r3_u(1) - r3_u(2))/(e3_u(1) - e3_u(2));
x_33_u = a(ind_110(3))-(ff4(ind_110(3))-1)/slope3_u;
e32_u = [a(1+405), x_33_u];
r32_u = [ff(1+405), 1];

e4_u = [a(ind_100(4)),a(ind_110(4))];
r4_u = [ff3(ind_100(4)),ff4(ind_110(4))];
slope4_u = (r4_u(1) - r4_u(2))/(e4_u(1) - e4_u(2));
x_34_u = a(ind_110(4))-(ff4(ind_95(4))-1)/slope4_u;
e42_u = [a(1+495), x_34_u];
r42_u = [ff(1+495), 1];

e5_u = [a(ind_100(5)),a(ind_110(5))];
r5_u = [ff3(ind_100(5)),ff4(ind_110(5))];
slope5_u = (r5_u(1) - r5_u(2))/(e5_u(1) - e5_u(2));
x_35_u = a(ind_110(5))-(ff4(ind_110(5))-1)/slope5_u;
e52_u = [a(1+575), x_35_u];
r52_u = [ff(1+575), 1];

e6_u = [a(ind_100(6)),a(ind_110(6))];
r6_u = [ff3(ind_100(6)),ff4(ind_110(6))];
slope6_u = (r6_u(1) - r6_u(2))/(e6_u(1) - e6_u(2));
x_36_u = a(ind_110(6))-(ff4(ind_110(6))-1)/slope6_u;
e62_u = [a(655), x_36_u];
r62_u = [ff(655), 1];

e7_u = [a(ind_100(7)),a(ind_110(7))];
r7_u = [ff3(ind_100(7)),ff4(ind_110(7))];
slope7_u = (r7_u(1) - r7_u(2))/(e7_u(1) - e7_u(2));
x_37_u = a(ind_110(7))-(ff4(ind_110(7))-1)/slope7_u;
e72_u = [a(739), x_37_u];
r72_u = [ff(739), 1];

e8_u = [a(ind_100(8)),a(ind_110(8))];
r8_u = [ff3(ind_100(8)),ff4(ind_110(8))];
```

```
slope8_u = (r8_u(1) - r8_u(2))/(e8_u(1) - e8_u(2));
x_38_u = a(ind_110(8))-(ff4(ind_110(8))-1)/slope8_u;
e82_u = [a(840), x_38_u];
r82_u = [ff(840), 1];

e9_u = [a(ind_100(9)),a(ind_110(9))];
r9_u = [ff3(ind_100(9)),ff4(ind_110(9))];
slope9_u = (r9_u(1) - r9_u(2))/(e9_u(1) - e9_u(2));
x_39_u = a(ind_110(9))-(ff4(ind_110(9))-1)/slope9_u;
e92_u = [a(945), x_39_u];
r92_u = [ff(945), 1];

e10_u = [a(ind_100(10)),a(ind_110(10))];
r10_u = [ff3(ind_100(10)),ff4(ind_110(10))];
slope10_u = (r10_u(1) - r10_u(2))/(e10_u(1) - e10_u(2));
x_310_u = a(ind_110(10))-(ff4(ind_110(10))-1)/slope10_u;
e102_u = [a(1005), x_310_u];
r102_u = [ff(1005), 1];

e11_u = [a(ind_100(11)),a(ind_110(11))];
r11_u = [ff3(ind_100(11)),ff4(ind_110(11))];
slope11_u = (r11_u(1) - r11_u(2))/(e11_u(1) - e11_u(2));
x_311_u = a(ind_110(11))-(ff4(ind_110(11))-1)/slope11_u;
e112_u = [a(1095), x_311_u];
r112_u = [ff(1095), 1];

e12_u = [a(ind_100(12)),a(ind_110(12))];
r12_u = [ff1(ind_100(12)),ff2(ind_110(12))];
slope12_u = (r12_u(1) - r12_u(2))/(e12_u(1) - e12_u(2));
x_312_u = a(ind_110(12))-(ff4(ind_110(12))-1)/slope12_u;
e122_u = [a(1170), x_312_u];
r122_u = [ff(1170), 1];

e13_u = [a(ind_100(13)),a(ind_110(13))];
r13_u = [ff3(ind_100(13)),ff4(ind_110(13))];
slope13_u = (r13_u(1) - r13_u(2))/(e13_u(1) - e13_u(2));
x_313_u = a(ind_110(13))-(ff4(ind_110(13))-1)/slope13_u;
e132_u = [a(1255), x_313_u];
r132_u = [ff(1255), 1];
```

```
e14_u = [a(ind_100(14)),a(ind_110(14))];
r14_u = [ff3(ind_100(14)),ff4(ind_110(14))];
slope14_u = (r14_u(1) - r14_u(2))/(e14_u(1) - e14_u(2));
x_314_u = a(ind_110(14))-(ff4(ind_110(14))-1)/slope14_u;
e142_u = [a(1385), x_314_u];
r142_u = [ff(1385), 1];

e15_u = [a(ind_100(15)),a(ind_110(15))];
r15_u = [ff3(ind_100(15)),ff4(ind_110(15))];
slope15_u = (r15_u(1) - r15_u(2))/(e15_u(1) - e15_u(2));
x_315_u = a(ind_110(15))-(ff4(ind_110(15))-1)/slope15_u;
e152_u = [a(1515), x_315_u];
r152_u = [ff(1515), 1];
%}

%% Displaying lines separating teeth
%
imshow(J);
hold on
plot(a,f,'color','green');
hold on
plot(e_12,r_12,'color','green');
hold on
plot(e22,r22,'color','green');
hold on
plot(e32,r32,'color','green');
hold on
plot(e42,r42,'color','green');
hold on
plot(e52,r52,'color','green');
hold on
plot(e62,r62,'color','green');
hold on
plot(e72,r72,'color','green');
hold on
plot(e82,r82,'color','green');
hold on
plot(e92,r92,'color','green');
```

```
hold on
plot(e102,r102,'color','green');
hold on
plot(e112,r112,'color','green');
hold on
plot(e122,r122,'color','green');
hold on
plot(e132,r132,'color','green');
hold on
plot(e142,r142,'color','green');
hold on
plot(e152,r152,'color','green');
hold on
plot(e_12_u,r_12_u,'color','green');
hold on
plot(e22_u,r22_u,'color','green');
hold on
plot(e32_u,r32_u,'color','green');
hold on
plot(e42_u,r42_u,'color','green');
hold on
plot(e52_u,r52_u,'color','green');
hold on
plot(e62_u,r62_u,'color','green');
hold on
plot(e72_u,r72_u,'color','green');
hold on
plot(e82_u,r82_u,'color','green');
hold on
plot(e92_u,r92_u,'color','green');
hold on
plot(e102_u,r102_u,'color','green');
hold on
plot(e112_u,r112_u,'color','green');
hold on
plot(e122_u,r122_u,'color','green');
hold on
plot(e132_u,r132_u,'color','green');
hold on
```

```
plot(e142_u,r142_u,'color','green');
hold on
plot(e152_u,r152_u,'color','green');

% Displaying two points defining gradient
%{
imshow(J);
hold on
plot(a,f);
hold on
%Upper Jaw
plot(e1_u,r1_u);
hold on
plot(e2_u,r2_u);
hold on
plot(e3_u,r3_u);
hold on
plot(e4_u,r4_u);
hold on
plot(e5_u,r5_u);
hold on
plot(e6_u,r6_u);
hold on
plot(e7_u,r7_u);
hold on
plot(e8_u,r8_u);
hold on
plot(e9_u,r9_u);
hold on
plot(e10_u,r10_u);
hold on
plot(e11_u,r11_u);
hold on
plot(e12_u,r12_u);
hold on
plot(e13_u,r13_u);
hold on
plot(e14_u,r14_u);
hold on
plot(e15_u,r15_u);

%Lower Jaw
hold on
plot(e1,r1);
hold on
plot(e2,r2);
hold on
plot(e3,r3);
hold on
plot(e4,r4);
hold on
plot(e5,r5);
hold on
plot(e6,r6);
hold on
plot(e7,r7);
hold on
plot(e8,r8);
hold on
plot(e9,r9);
hold on
plot(e10,r10);
hold on
plot(e11,r11);
hold on
plot(e12,r12);
hold on
plot(e13,r13);
hold on
plot(e14,r14);
hold on
plot(e15,r15);
%}

%% Calculating angles of teeth

% Upper Jaw
```

```
theta1 = radtodeg(atan(0-(slope1_u+slope2_u)/2));
theta2 = radtodeg(atan(0-(slope2_u+slope3_u)/2));
theta3 = radtodeg(atan(0-(slope3_u+slope4_u)/2));
theta4 = radtodeg(atan(0-slope4_u));
theta5 = radtodeg(pi+atan(0-(slope5_u)));
theta6 = radtodeg(atan(0-(slope6_u+slope7_u)/2));
theta7 = radtodeg(atan(0-(slope7_u)));
theta8 = radtodeg(pi+atan(0-(slope8_u+slope9_u)/2));
theta9 = radtodeg(pi+atan(0-(slope9_u+pi/2)/2));
theta10 = radtodeg(atan(0-(slope10_u+slope11_u)/2));
theta11 = radtodeg(atan(0-slope11_u));
theta12 = radtodeg(pi+atan(0-slope12_u));
theta13 = radtodeg(atan(0-slope13_u));
theta14 = radtodeg(pi+atan(0-slope14_u));


% Lower Jaw

theta28 = radtodeg(atan(0-(slope1+slope2)/2));
theta27 = radtodeg(atan(0-(slope2+slope3)/2));
theta26 = radtodeg(atan(0-(slope3+slope4)/2));
theta25 = radtodeg(atan(0-(slope4+slope5)/2));
theta24 = radtodeg(atan(0-(slope5+pi/2)/2));
theta23 = radtodeg(atan((pi/2+pi/2)/2));
theta22 = radtodeg(atan((pi/2+pi/2)/2));
theta21 = radtodeg(atan(0-(slope8+slope9)/2));
theta20 = radtodeg(atan(0-(slope9+slope10)/2));
theta19 = radtodeg(pi+atan(0-(slope11)));
theta18 = radtodeg(pi+atan(0-(slope11+slope12)/2));
theta17 = radtodeg(pi+atan(0-slope12));
theta16 = radtodeg(pi+atan(0-slope13));
theta15 = radtodeg(pi+atan(0-slope14));

upper_jaw = [theta1; theta2; theta3; theta4; theta5; theta6;
    theta7;theta8;theta9;theta10;theta11;theta12;theta13;theta14];
lower_jaw = [theta15;theta16;theta17;theta18;theta19;theta20;
    theta21;theta22;theta23;theta24;theta25;theta26;theta27;theta28];

% Output
```

```
angular_data = [upper_jaw, lower_jaw];

end
```