Anyone who has never made a mistake has never tried anything new. . ----------***Albert Einstein***

Dedicated

to my

Maa & Baba

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts. An understanding of the study like this, is never the outcome of the effort of a single person, rather it bears the imprint of a number of people who directly or indirectly helped me in partial fulfilment of this project. I would be failing in my duty if I don't say a word of thanks to all those people.

In the first place I would like to record my gratitude to **Dr. H K Sardana**, Deputy Director, Scientist G, DU-2, CSIO, for his guidance, valuable suggestions, advice, guidance and encouragement to working towards this thesis. Above all and most needed, he provided me unflinching encouragement and support in various ways. His truly scientist intuition has made him a constant oasis of ideas and passion in science, which exceptionally enrich my growth as a student. I gratefully acknowledge my indebtness to him for his constant inspiration, painstaking effort and illuminative guidance, throughout the course of this study. I will be grateful to him for introducing me in the field of Image processing.

I would like to thank and express my deep sense of gratitude to my esteemed teacher and a great gracious advisor **Dr. Amod Kumar,** Scientist G, DU-2, CSIO, whose involvement worked meticulously all through with the special vigilance and zeal. His involvement with his enriched knowledge has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

I would like to thank Rahele Kafieh of Isfahan University of medicine, Hezar jareeb St, Isfahan, Iran for continuously supporting me during my work period.

I would like to thank Prof. Ghassan Hamarneh (Simon Fraser University, Canada)Yi-Hsien Lin , (National Cheng Kung University, Taiwan), Lihong Ma (South China Univ.of Tech,China) for clearing my doubts and constantly supporting me during the work.

I must place on records the encouragement and moral support extended by the project assistant Er. Asish Jain, CSIO for his constant help and support throughout my thesis period.

I am also grateful to the staff of All India Institute of Medical Science and Post graduate institute of education and medical research ,India for providing digitized x-rays.

I find it difficult to verbalize my deepest sense of indebtedness to my parents and friends for their boundless love and support, which has always been a source of inspiration that kept me going to complete this venture.

TANMOY MONDAL

19th JUN, 2009

ABSTRACT

As a routine clinical procedure in orthodontics there are many geometrical analysis performed on the lateral x-ray images of the head (called cephalogram) resulting in an appropriate treatment plan. These geometric analyses mostly involve the manual determination of some specific anatomical locations on the cephalogram, which are called landmarks. These landmarks are then used for measurement of the dimensions and angles between the bony structures within the head. This method was first introduced by Broadbent in 1931 and is known as Cephalometric Analysis. Hence, landmark localization is an important step in cephalometric analysis .The most common method for cephalometric analysis is by manually identifying landmarks, measuring distance and angles between landmark locations. Another approach is computer aided; landmarks are located manually while these locations are digitized into a computer system and the system then completes cephalometric analysis. The third approach is completely automated. The cephalometric radiograph is scanned; the computer automatically locates landmarks and performs cephalometric analysis. Thus the process of automatic landmark identification has the potential to increase the accuracy, provide more efficient use of clinical time, and improves the ability to correctly diagnose orthodontic problems.

In this work, we have attempted to develop a novel algorithm for automatic tracing of anatomical structure of cephalogram images and locating the landmarks. Dividing the scanned cephalogram into different rectangular regions that then applying the edge detection algorithm on the cropped one, then placing the desired edge on the original image. A template-matching algorithm is used to detect the rectangular regions. To reduce processing time and eliminate risk of false alarms, pattern matching is applied only in the zones determined by the prior knowledge of human skull and the anatomical structures locations.

In the second approach the combination of image processing technique and statistical model is proposed for feature point localisation. On cephalogram human experts locate landmarks by hand and the locations of their located co-ordinates are stored in a database. For each image, principal component analysis (PCA) is applied on the located landmarks for statistically characterize its shape variation and grey profile of each feature point. For an input sample, the reference landmarks are recognised by edge detection and template matching technique, and used to divide the input shape. Then the feature points of each input shape are located by a modified active shape model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

This chapter covers an overview of cephalometric analysis followed by introduction to landmark localization for cephalometric analysis and the methods of landmark localization.

***INTRODUCTION*:** Cephalometric analysis is an important tool in orthodontic diagnosis, treatment planning, and evaluation of treatment results and prediction of growth. Hand measurements are time consuming and there is always a risk of human errors. Rapid advances in computer science have led to its wide application in Cephalometry, therefore in recent years digital cephalometric analysis is gaining popularity in orthodontic practices. The use of modern cephalometric software requires import of digital cephalograms or digital capture of analogue data: scanning and digital photography. Moyers [1] was stating that cephalometrics is a radiographic technique for abstracting the human head into a geometric scheme. Cephalometric analysis is currently used in order to evaluate skeletal and dental development, to identify malloclusion, to plan the treatment and to evaluate therapy results [2, 3]. Some geometrical analysis is performed on the lateral x-ray images of the head (called cephalogram) for approximating the treatment plan. On the base of this geometrical analysis some manual determination of some specific anatomical locations on the image which is called landmarks. On the base of this landmark measurement of the dimensions and angle between the bony structures within the head is performed. Eighty years ago a method for standardized head radiography was introduced which turned out to be a tremendous advance in the measurement of the growth of the head and face. In 1931 Broadbent in the USA and Hofrath in Germany published on the methodology of obtaining standardized cephalometric head films. But it was until 1948 that the first cephalometric analysis in the USA was published by Downs, who introduced a practical cephalometric analysis for diagnostic purposes. Downs' has been the basis for most methods used at present. Nowadays, analyzing cephalometric radiographs is one of the most commonly used clinical procedures in any orthodontic office. The 2002 JCO study of orthodontic diagnosis and treatment procedures among orthodontists in the USA (response rate 9%), previously conducted in 1986, 1990, and 1996, showed that the most commonly used analysis was the Steiner analysis (in 45.1% of the practices) and its relative popularity as compared to other analyses remained about the same over the years. A survey among all orthodontists working in the Netherlands (response rate 78%) showed similar findings. The most commonly used analysis also was the Steiner analysis, used by 58% of the Dutch orthodontists, followed by the Downs analysis that was used

by 22%. This study also showed that most orthodontists used more than one cephalometric analysis for diagnosis and treatment planning.

Analysis available:-

Downs analysis, Ricketts analysis, Steiner analysis, Tweed analysis, Wits analysis

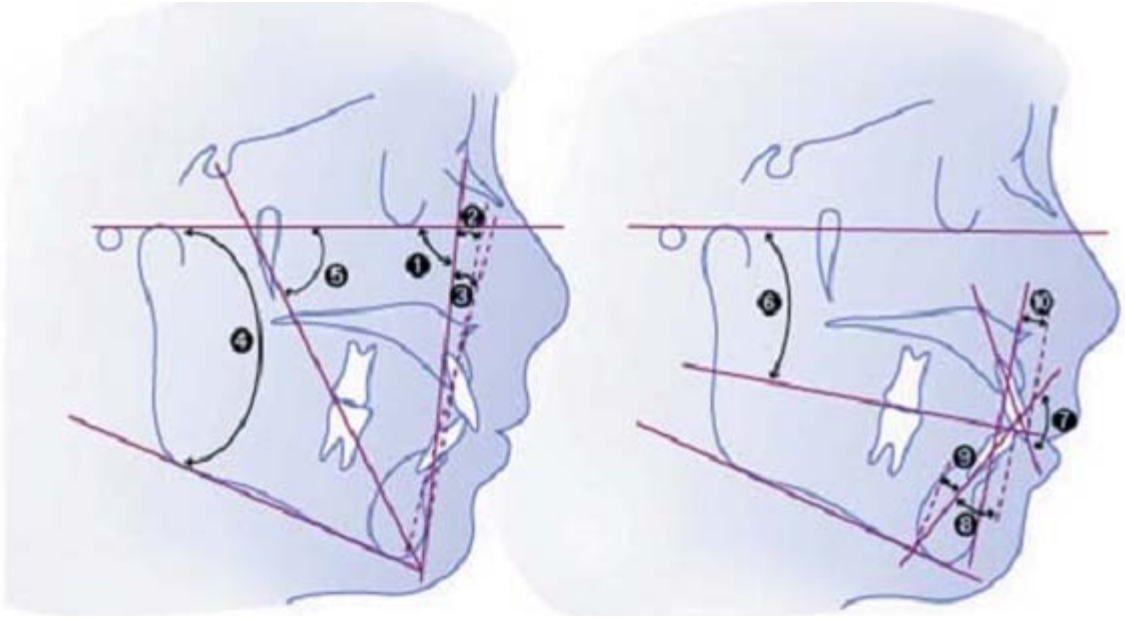


Figure 1.1: Downs's analysis with the reference points identified. 1, Facial plane. 2, Convexity. 3, A-B plane. 4, Mandibular plane. 5, Y axis. 6, Occlusal plane. 7, Interincisal angle. 8, T to Occlusal plane. 9, T to mandibular plane. 10, I to A-P plane.

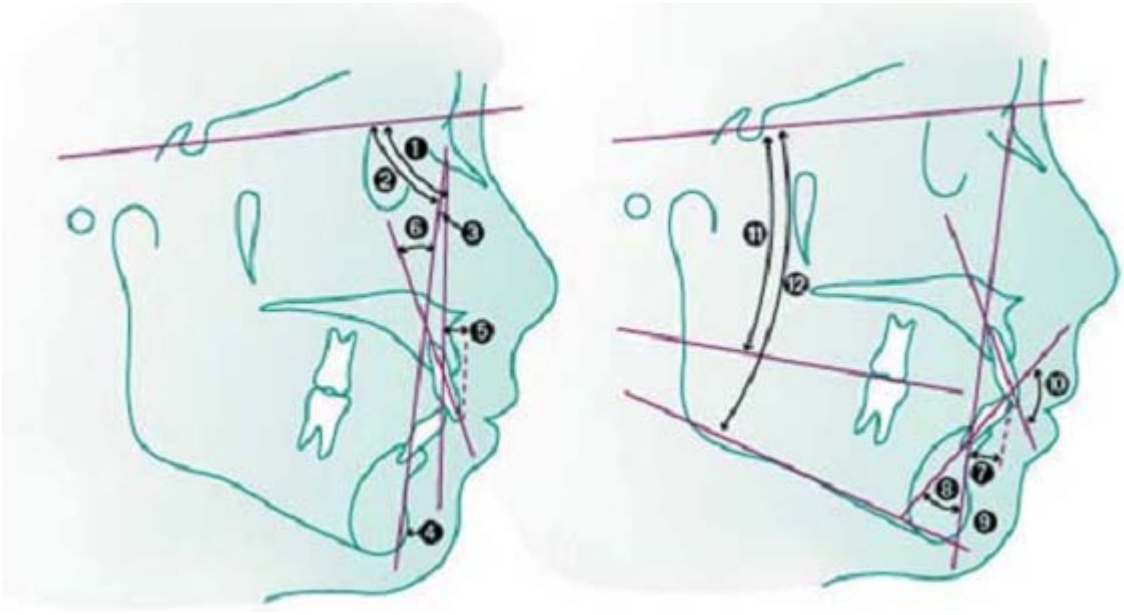


Figure 1.2 : Steiner's analysis with the reference points identified. 1. SNA; 2. SNB; 3. ANB; 4. SND; 5. I TO NA (mm); 6. I to NA (angle); 7. I to NB (MM); 8. I to NB (angle); 9. Po to NB (MM); 10. Interincisal; 11. Occlusal to SN; 12. GoGn to SN.

According to the Steiner's analysis: ANB (A point, Nasion, B point)- indicates whether the skeletal relationship between the maxilla and mandible is a normal (skeletal class I relationship (+ 2 degrees), a skeletal Class II (+4 degrees or more), or skeletal class III (0 or negative). SNA (sella, Nasion, A point)- indicates whether or not the maxilla is normal, prognathic, or retrognathic. SNB (sella, Nasion, B point)- indicates whether or not the mandible is normal, prognathic, or retrognathic. SNA and SNB is important to determine what type of intervention (on maxilla, mandible or both) is appropriate. These angles, however are influenced also by the vertical height of the face and a possible abnormal positioning of Nasion.

There are three possible methods to perform a cephalometric analysis. One method is by manually placing a sheet of acetate over the cephalogram, tracing silent features, identifying landmarks, and measuring distance and angle between landmark locations. Another approach is computer aided; landmarks are located manually while these locations are digitized into a computer system the computer then complete cephalometric analysis. The third approach is completely automated. The cephalometric radiograph is scanned into the computer; the computer automatically locates landmarks and performs the cephalometric analysis. Thus the process of automatic landmark identification has the potential to increase the accuracy, provide more efficient use of clinician time, and improve the ability to correctly diagnose orthodontic problems.

CHAPTER II:

CEPHALOMETRY

2.1 CEPHALOMETRIC ANALYSIS

As a routine clinical procedure in orthodontics there are many geometrical analysis performed on the lateral X-ray images of the head called cephalogram (Figure 2.1), resulting in an appropriate treatment plan.

Cephalometric analysis is the scientific measurement of the dimensions of head. Current practice uses a two dimensional radiographic image on film. The patient is placed in a cephalostat (Figure 2.2) with their head oriented at 90 degree to X-ray beam at a distance of 5 ft. from the tube. This is a standard under which all cephalometric radiographs are taken worldwide. In orthodontics, a cephalometric analysis performed on a lateral skull X-ray, or cephalogram, reduces anatomic structures to landmark points supposed to indicate shapes and relative locations of curves. Analysis of the cephalogram images and measurement of parameters based on the detected feature points - landmarks - play an essential role in diagnosis and treatment planning by orthodontists. Lines are constructed from the detected landmarks, and the subsequent measurement of various angular and linear parameters generates an evaluation of the facial skeleton and teeth for the orthodontist to assess.

The conventional method of locating landmarks & drawing the anatomical structure depends on manual tracing of the radiographic image. The process is tedious, operator dependent, time consuming and subject to systematic error. An automated system would eliminate human subjectivity and hence play an important part in producing repeatable results. Investigations into automatic landmark recognition using different image processing techniques have been undertaken for several years. The task is a difficult one due to the complexity and variability of the images.

2.2 APPROACHES TO PERFORM CEPHALOMETRIC ANALYSIS

Cephalometry the scientific measurement of the dimensions of head usually on X-ray images Localization of cephalometric landmarks is essential to craniofacial surgery & is widely used in Orthodontics, orthopaedics, and other areas of oral and maxillofacial surgery to assess and predict Craniofacial growth, plan treatment, evaluates curative effect, and compares different cases. It

Detects feature points for further parameter measurement. Conventional methods of landmark Location include manually tracing, automatic recognition and man-machine interactive conversation, but all these kinds have some limitations. Manual tracing depends on orthodontists' experiences, which is time consuming, tedious and unrepeatable, subject to human mistakes as well On the other hand, automatic recognition is the most difficult task due to the complexity Caused by skeleton overlapping and no uniform illumination in X-ray cephalogram.

There are two important ways to obtain the cephalometric analysis:

a) Landmark localization and *b*) craniofacial anatomical structure tracing.

Generally, cephalometric analysis is based on a set of agreed-upon feature points named craniofacial Landmarks. In orthodontics there are 90 landmarks, 30 of which are commonly used. Once the Landmarks are located, the measurement and analysis of various angular and linear parameters can be performed. In manual land marking, orthodontist's usually first trace out the craniofacial structure contours on X-ray images, and then extract landmarks from corners, line intersections and other geometrical line shapes. This process, averagely costing an experienced orthodontist 25–30 min on each case is tedious, time consuming. Furthermore, the accuracy will inevitably influenced by the fatigue level of human. Computerized land marking, therefore, is an important subject. Early works, being called *hand-crafted* algorithms, usually locate landmarks based on edge detection techniques. Its basic idea is to simulate manual procedure. After extracting relevant edges, algorithm with prior knowledge on typical shapes of craniofacial structures are used to track lines in a predetermined order of detected lines. Positions of the landmarks are determined based on a set of predefined geometrical properties of lines, intersections and exterior boundaries. The following figure (2.1) shows this kind of analysis base on the Down's methodology known as Down's analysis.

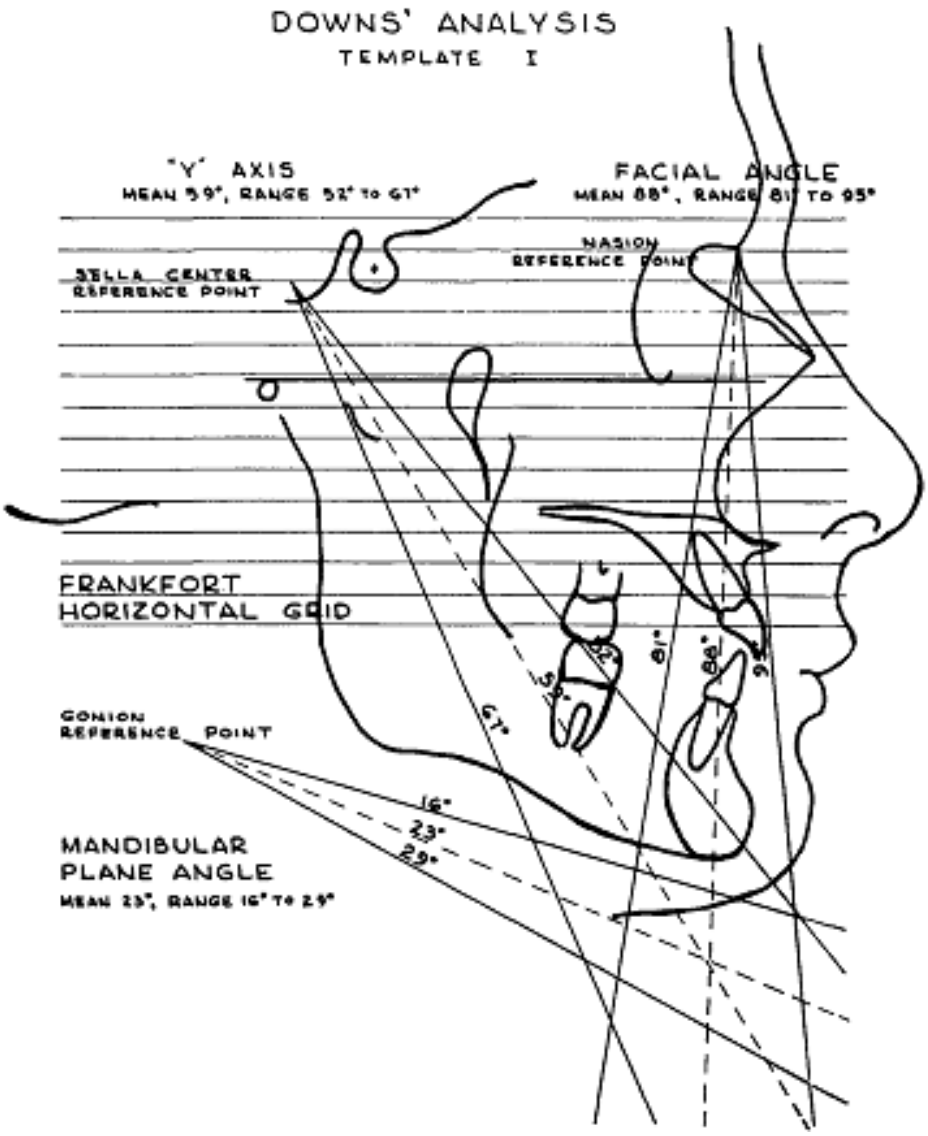


Fig 2.1: Template I oriented over head plate for determination of Facial Angle. note nasion reference point of the template on 'nasion of the x-ray; and the Frankfort Horizontal Grid lines of the template parallel to the Frankfort Horizontal Plane of the x-ray. The Facial Angle is read from the location of pogonion with reference to the lines radiating from the nasion reference point. The dashed centre line indicates the mean Facial Angle of 88°.The solid lines on either side delineate normal range. The actual numerical value of the Facial Angle in this case is 91.5°.

APPROACHES TO PERFORM CEPHALOMETRIC ANALYSIS

2.2 MANUAL ANALYSIS

2.2.1 TRACING TECHNIQUE

Tracing of cephalogram should be undertaken in a darkened room. Doctors use good quality tracing paper securely taped to radiograph (along the top edge of the tracing paper, directly to the radiograph. A sharp (HB) pencil is used to outline the following:

1. Soft tissue profile of the face (forehead to chin)
2. Sella turcica
3. Frontal bone and nasal bone
4. Orbital floor

Tracing should now look like this:

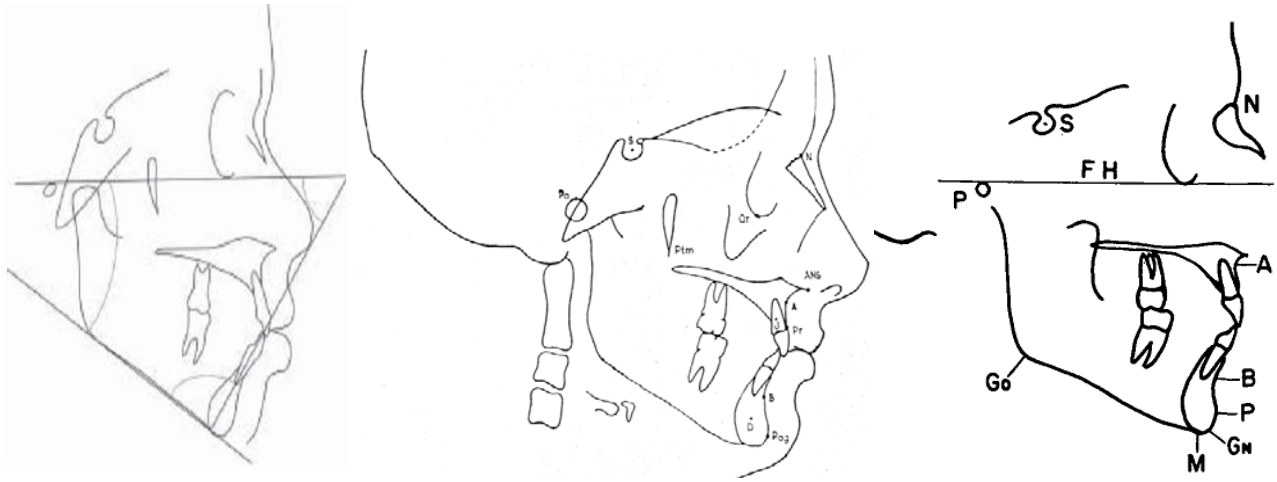


Figure 2.2: Tracing of cephalogram

After this doctors localize landmarks and draw the different lines and planes on the basis of these detected landmarks. Various angles and distances are then calculated between these lines and then tracing should now look like Figure 2.5. Figure below (2.3) shows a twelve year old patient. She was seeking for orthodontic treatment. The clinical examination shows that it is a case of angles class I type cravding. Case of class I type malocclusion with increased over jet and overbite. It is a class 3 ectopic eruption class 4 vertically impacted. Orinasal breathing pattern and typical yellowing pattern with lower middle shift to the left by 2 mm.

Pre-treatment Extra-Oral photographs

Front | Rt Lateral | 45° View

Sristi 12/F
Ortho. No:17887/04
Date : 1.08.04

Sristi 12/F
Ortho. No:17887/04
Date : 1.08.04

Figure 2.3:(a) Pre-treatment intra - oral Photographs

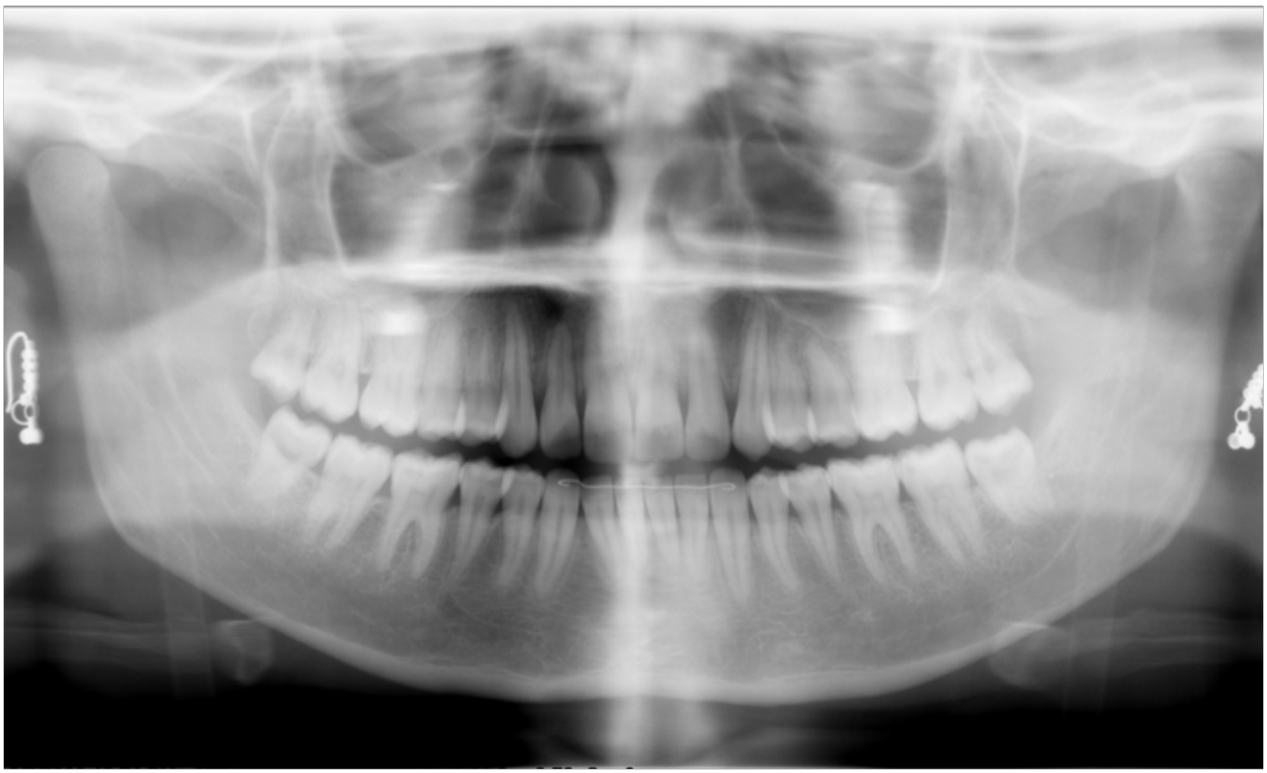


Figure 2.3: (b) Shows Pre-treatment panoramic photograph of a patient

Mid-treatment Extra-Oral photographs

Sristi 13/F
Ortho. No:17887/04
Date : 20.10.05

Mid - treatment Intra-oral photographs (Pre-MPA IV)

Sristi 13/F
Ortho. No:17887/04
Date : 20.10.05

Mid – treatment Intra – Oral Photographs (With MPA IV)

Sristi 13/F
Ortho. No:17887/04
Date : 02.03.06

Mid – treatment Intra – Oral Photographs (5 months of MPA IV)

Sristi 13/F
Ortho. No:17887/04
Date : 02.03.06

Mid – treatment Intra – Oral Photographs (20 months of treatment)

Fig 2.4: Shows patient's oral structure at the middle of the treatment

2.2.2 COMPUTER-AIDED ANALYSIS

It is both time-consuming and somewhat inaccurate technique of manually analyzing a cephalometric radiograph by measuring spatial relationships between cephalometric landmarks. This time pressures in the clinical environment can contribute to decreased reliability. Currently, errors in manual cephalometric analysis can be attributed to errors in reproducibility and validity. Recently a computer-assisted method of cephalometric analysis has been developed, it is a semiautomatic approach. This process requires manual landmark identification and digitization along with computerized measurement of the landmark relationships. This approach provides for comprehensive measurements in less time, but it is still time-consuming and error-prone. Currently, several commercially available systems can perform basic cephalometric analysis tasks.

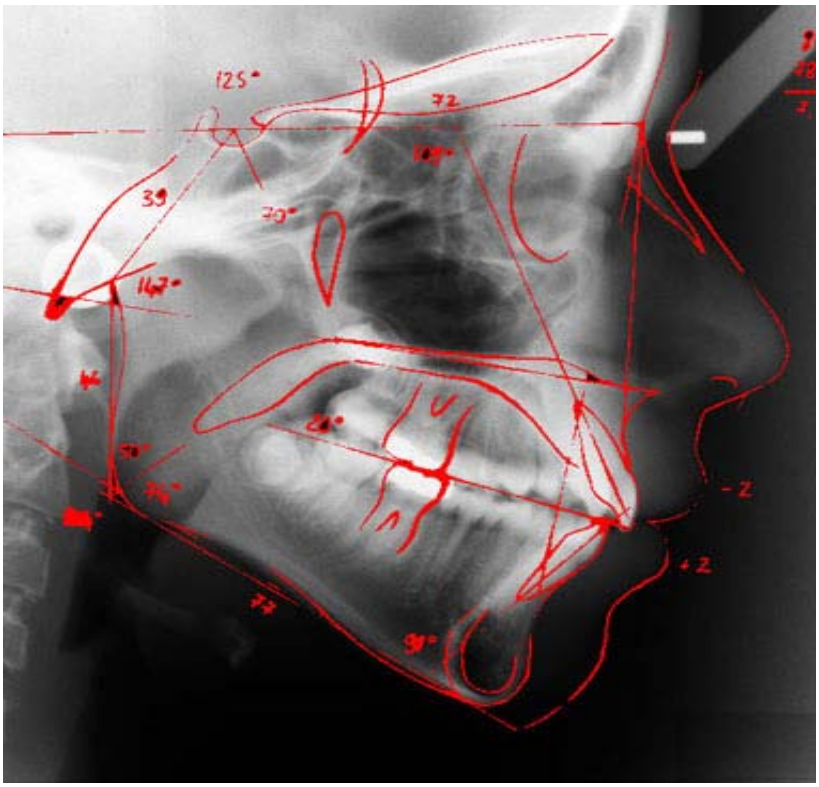


Figure 2.5: Shows example of manual cephalometric analysis.

Other systems digitize landmark locations on a digitizing pad. In either case a computer algorithm performs a cephalometric analysis by calculating distances and angles between landmark locations. In addition, the algorithm connects these landmarks with line segments to produce a tracing. Some systems are capable of moving the tissues to simulate treatment effects, growth effects, and surgical prediction. Finally, some of these systems also are able to produce a time

series of images using landmark locations, not superimposition contours, to register images. Generally, these systems do not save time, are expensive, and require technical training. The accuracy of these computer-aided programs has been demonstrated to be similar to that of manual digitization, and because manual landmark identification programs require subjective user point identification, they are limited in scope. In addition, the numbers of landmarks required are high; this tends to negate any time saved using this method. Although the analysis uses a computer, the process of manual point digitization can be time-consuming and error-prone.

Variations in image acquisition can result from beam alignment and repositioning, exposure parameter variations, and film processing variations. Limitations of human visual performance may result in cephalometric landmark identification errors. These errors may be expressed as either Inter observer or intra observer variation. Inter observer variation also may be caused by variations in training and experience, or by the subjective nature of landmark identification. Furthermore, intra observer variation may result from variations in lighting and image orientation. It is generally accepted that land- marking errors should be less than 1 mm to 2 mm. This rule of thumb, however, has not been proven in research studies. The mean estimating error of landmark identification has been reported to be 1.26 mm in one study. This study also showed that manual landmark identification errors varied significantly depending on the landmark and the observer. Landmarks currently are defined using subjective criteria rather than mathematically rigorous specifications.

Landmarks, for the most part, are described as extreme of a tissue component with respect to a spatial coordinate. For example, pogonion is the most anterior point of the chin. Thus the orientation of the patient's head affects the location of pogonion. Other landmarks that are affected by image orientation include A point, B point, menton, and gonion. Measurement of distances and angles between landmark locations is defined by the limitations of the measurement devices (rulers and protractors), as well as the limitations of human visual performance. Human errors can occur both in recording of the measurement or in the use of the measurement devices. In most cases, landmarks are not labelled before measurement because of time constraints, causing additional errors. The time required for manual analysis depends on how comprehensive the measurements are. Some clinicians make no measurements at all; they hold the radiograph up to the light and get a "clinical impression" of the skeletal and dental pattern of the patient. Other clinicians make extensive, time-consuming measurements, and some clinicians direct a staff member (no orthodontist) to prepare the analysis. In each case the technique is inefficient, time

consuming, or prone to error at several steps. Following figure (2.6) shows semiautomatic computer aided approach for manually tracing the anatomical structure.

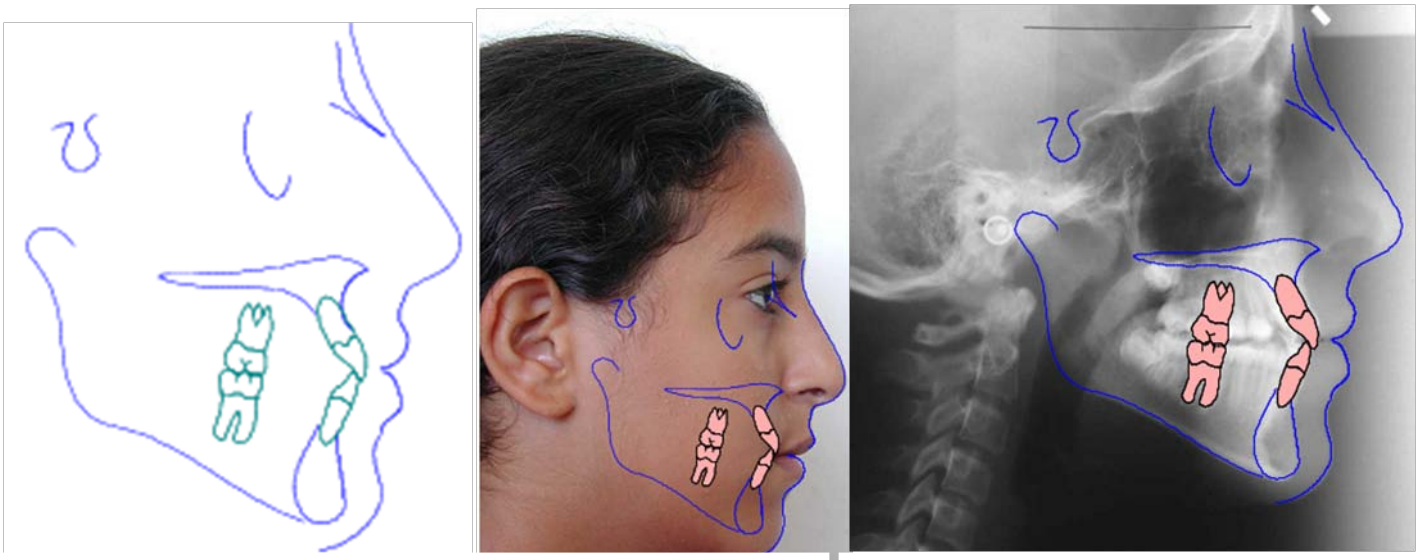


Fig 2.6 (a): Tracing technique (b)Profile Overlay (c)Image Overlay

2.2.3 AUTOMATIC LANDMARK IDENTIFICATION AND STRUCTURE TRACING

Anatomical structure tracing and landmark identification on cephalograms is a significant way to obtain cephalometric analysis. The ways for computerized cephalometric analysis are divided in two main categories, manual and automatic approaches. The manual approach is limited in accuracy and repeatability. Here we have attempted to develop and test a novel method for automatic localization of craniofacial structure. This approach to cephalometric analysis is completely automated. The cephalometric image is scanned into a computer and landmark identification, structure tracing and cephalometric analysis are automated. The process has the potential to increase accuracy, provide more efficient use of clinician's time, and improve the ability to correctly diagnose orthodontic problems. Additionally, this process may provide mathematical descriptions of landmark locations that could be applied to new ways of evaluating cephalometric radiographs to derive clinically important information.

The following figures demonstrates location of desired craniofacial structures and landmarks on the cephalometric images, detection of those is our desired objective.

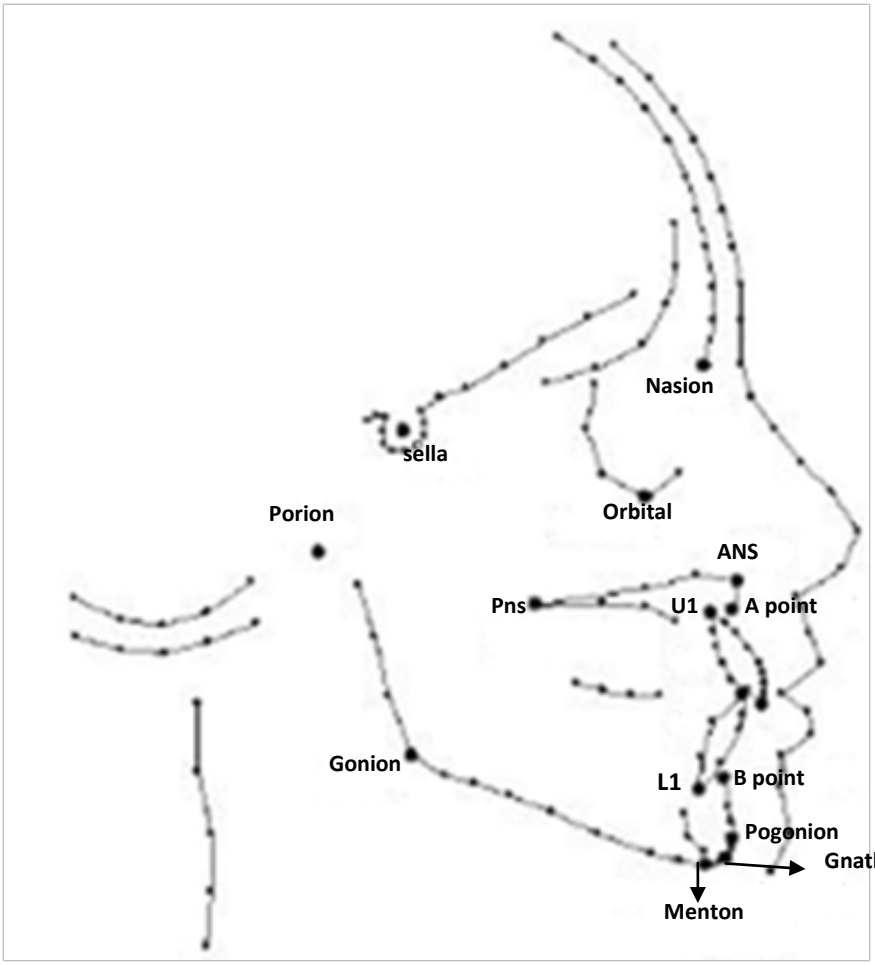


Fig 2.7: Demonstrates the definition of different desired landmarks required for cephalometric analysis

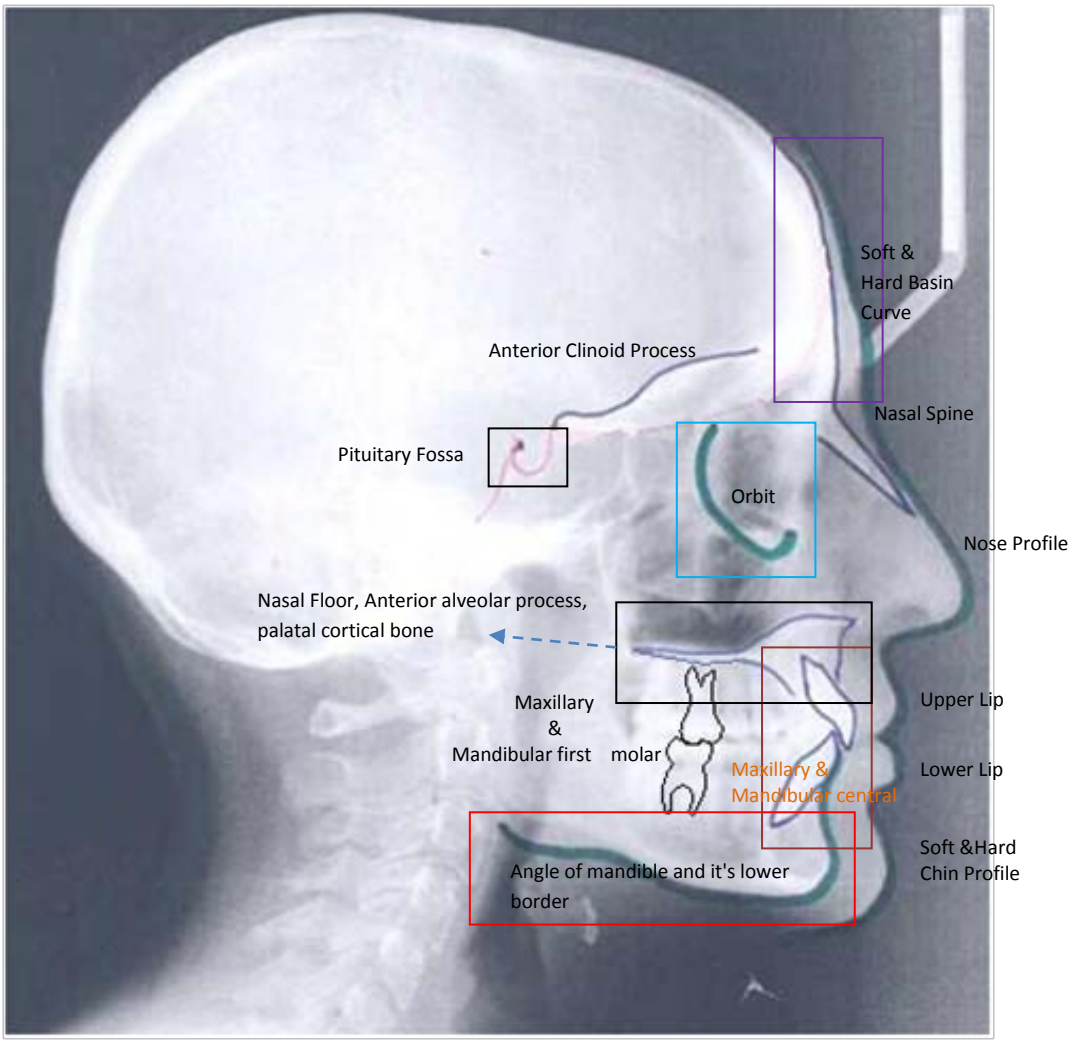


Fig 2.8 : Demonstrates the definition of different desired craniofacial structures required for cephalometric analysis

2.3 THESIS ORGANIZATION

In chapter 2, we present an overview of the advancements made in the field of cephalometric analysis. It briefly presents the journey of cephalometric analysis from simple manual localization of landmarks to semi-automatic cephalometric and gives some glimpse of what the cephalometric analysis basically is followed by introduction to automatic landmark localization for cephalometric analysis. In chapter 3, we present extensive literature survey on previous work done for automatic anatomical structure tracing on cephalogram images for cephalometric analysis. Chapter 4 deals with the “Automated 2-D cephalometric analysis on X-ray images by the edge detection approach”, in this approach we are basically detect the edges from the image and track our desired edge and extract it than show it on our main image. Then the results are shown, made on the base of this algorithm.

Chapter 5 deals with the "Automated cephalometric analysis by model based approach ". In this chapter an approach is made by active shape modelling and grey level statics extraction of the model point. Then results are shown is made on the base of this algorithm.

Chapter 6 deals with the common discussion on the base of two algorithmic approaches, Computational issue of the two algorithms and the common discussion of the two approaches. This chapter also describes the future approaches which can be used to improve the result into a better destination.

2.4 MATERIALS: It is really praiseworthy to have a randomly selected data set without any judgement of their quality, sex and age; to ascertain that the proposed methodology is an efficient approach for obtaining desired result. In this research 55 pre-treatment cephalograms were used and will be accessible on request for comparison by other researchers. In this full dataset a set of 30 images (say dataset1) were obtained by "VIDAR VXR-12" scanner with a resolution of 300 dpi (1 pixel=0.084 mm) and roughly sized to 2400 × 3000 pixels in DICOM format, and the set of 25 images (say dataset2) were obtained by "Strato 2000 Full Digital Dental Panoramic System" with a resolution of 194 dpi (1 pixel = 0.14 mm) and roughly sized to 1537 × 1171 pixels in JPEG format.

Sixty three pre treatment cephalograms were used for the model based approach which was collected by Rahele Kafieh [21], she collected this images and the database of the location of 114 landmarks for sixty three images from Hutton [12], which is accessible for comparison by other researchers.

Basically the experiment is performed by the drop one out method on this 63 images and its database which contains positions of one hundred forty feature points (including the sixteen landmarks) are used for training purpose for the model based approach.

CHAPTER III

LITERATURE SURVEY

This chapter presents the work done by other researchers to solve the problem of automatic landmark localization for cephalometric analysis and reviews the existing systems for cephalometric analysis

3.1 PREVIOUS WORK

Cephalometry was first introduced by Broadbent (1931) and subsequently revolutionized the analysis of malocclusion and the underlying skeletal structures. Lateral cephalograms may be traced manually but now the approach with computers has been introduced. It is widely acknowledged that both manual and computerized methods are time-consuming and error prone (Baumrind and Frantz, 1971; Broch et al., 1981). In addition, the process is open to considerable subjectivity. The associated errors were classified by Baumrind and Frantz (1971) as mechanical, projective or due to problems in landmark identification.

The greatest error lies in landmark identification and this, in turn, is related to the quality of the radiograph. Richardson (1981) compared digitization using an electro-mechanical device and manual tracing and found the former to be slightly more reproducible. There have been previous attempts to automate cephalometric analysis with the aims of reducing the time required to obtain an analysis, improving the accuracy of landmark identification and reducing the errors due to clinician subjectivity. The bid to establish such a system has become typical once more with the move towards digital imaging and the development of “film-free” hospitals. When on-screen digital images replace physical radiographs, as is already happening, it will no longer be possible to manually trace cephalograms as before and landmark annotation using a computer will become standard. A system that can automatically identify landmarks and produce the required measurements would be of immense benefit.

Some of the earliest work on automating cephalometric analysis was published by Levy-Mandel et al. in 1986 [5]. They tracked edges in the image to locate landmarks on structures with well-defined outlines, such as the lower border of the mandible. A prior knowledge of the typical shape of the important edges was encoded in algorithms that followed the boundaries of different structures. Each algorithm was designed to find a specific structure, this approach we term ‘hand-crafted algorithms.’ Their system was tested on two high-quality cephalograms scanned at 256 x

256 pixels with 256 grey levels. Only landmarks that lay on or near to edges in the image could be located. They reported that 23 out of 36 landmarks could be determined on a good quality image but gave no evaluation of the landmark accuracy as compared with an “expert.”

Parthasarathy et al. (1989) [6] presented a similar scheme, again using hand-crafted algorithms. They used 480 x 512 pixel images and created a four-level image pyramid for improving the efficiency of their search. No indication of the number of grey levels was given. Their system was tested on five cephalograms of varying quality. They compared the accuracy of their system to the landmarks as placed by two experts. Of nine landmarks, on average 18 percent were located to within 1 mm, 58 percent within 2 mm and 100 percent within 5 mm. It has been suggested that an error of 2 mm in landmark placement is acceptable (Rakosi, 1982) but that an accuracy of 1 mm is desirable (Forsyth et al., 1996). Tong et al. (1990) presented an extension to the work of Parthasarathy. Their system looked for 17 landmarks, not including any of the seven of the earlier work. They indicated that their results could be combined to yield a full cephalometric analysis. If both systems were used together, as was intended, on the five cephalograms on average 40 percent of 26 landmarks would be located to within 1 mm, 70 percent to within 2 mm and 95 percent to within 5 mm.

Davis and Taylor [7] (1991) described how hand-crafted algorithms could be integrated in a blackboard architecture, allowing back-tracking in the face of contradictions. A formal evaluation by Forsyth and Davis (1996) used 10 cephalograms scanned at 512 x 512 pixels with 64 grey levels. It excluded radiographs where un-erupted teeth were overlying the apices of the incisors. On average 63 percent of 19 landmarks were located to within 1 mm and 74 percent to within 2 mm. In all these earlier works it was not stated whether the hand-crafting of the algorithms was performed with the testing images unseen. Indeed, it was suggested by Rudolph et al. (1998) [8] that the algorithms were tested on the same images that were used to create them. To evaluate an image understanding algorithm it must be shown that it will perform acceptably on new images, not just those that have been used for designing the method. Without demonstrating that an approach can be generalized from the training set to a test set, a study is of little use. This flaw in scientific method means that direct comparison of the results of this study with those of the studies mentioned above is perhaps inadvisable. Each of the studies above used some algorithmic encoding of anatomical knowledge to locate landmarks on or near to edges in the image. While this approach gives good results for landmarks that are well-defined by strong edges, it tends to be unreliable in the face of image artefacts, variable image quality and structure changes such as malocclusions. It is not clear how such systems can be made more robust to the

presence of this variation without implementing ever more complex hand-crafted rules, together with error-checking and back-tracking mechanisms. The approach is intrinsically flawed with respect to these variations.

An alternative scheme, suggested by Cardillo and Sid-Ahmed (1994),[9] used sub-image matching on hand-selected features from a training set of 40 images. Their images were scanned at 512 x 480 pixels with 256 grey levels. Seventy-six percent of their 20 landmarks were located to within 2 mm. They did not specify how many were within 1 mm.

Rudolph et al. (1998) [8] described the use of spatial spectroscopy to characterize the grey-level appearance around each landmark from a training set of hand-landmarked images. They used a “drop-one-out” scheme to enable testing of 14 images using the other 13 as the training set. However, they used images that were just 64 x 64 pixels, to make their scheme computationally feasible. They compared their system's performance with that of an expert using images of the same resolution. They reported that no statistical difference could be found between the manual error and the error of their automated system. For comparison, this implied that 100 percent of the landmarks were located to within 4 mm. They suggested that as the resolution increased, the landmarks would correspondingly become more accurate but this remains to be shown. Most recently,

Chen et. al. (1999)[10] used neural networks together with genetic algorithms to search for sub-images that contained each of the cephalometric landmarks. However, they did not report on the accuracy of the landmark placement. These three later methods used essentially the same approach; generating a model of the grey-levels around each point from a training set and then matching this model to a new image to locate the points of interest. While this approach does not depend on strong, continuous edges, it does rely on the image appearance around each landmark being reproducible across all cephalograms seen. Around the upper incisor tip, for example, the grey-level appearance may vary greatly between different malocclusions.

Active shape models (ASMs) are described in Cootes et al. (1995) [11] and were first reported by Hill et.al. (1992), they used a model of the spatial relationships between the important structures - a template - to help search the image for features of interest. The key5innovation is that the variation in shape is modelled, enabling the synthesis of plausible new examples of the structures seen. By matching the deformable template to the structure in a particular image not only can the landmarks of interest be located but a complete atlas of all features can also be given.

The approach has been used to tackle other image processing problems within the field of dentistry (Hutton et al., 1999)[12].

The ASM approach uses both a local model of grey-level appearance and a global model of the spatial relationship between the points that define the target. The global model is not hand-crafted as in the earlier studies but is learned directly from the training set. The grey-level model is statistically derived from the images in the training set, as with Cardillo and Sid-Ahmed (1994), Rudolph et al. (1998) and Chen et al. (1999). Active shape model ASM) [11] was first employed to land marking in Hutton's study [12]. The algorithm was tested on 63 images, and 55% of 16 landmarks were within the range of 2 mm. Romaniuk [13],[14] used both linear and nonlinear models to locate landmarks with satisfactory accuracy. EI-Feghi [15], [16] demonstrated their methodology by the use of neural network to identify 25 most commonly used landmarks.

Chakrabartty [17] applied support vector machines (SVM) to detect 16 landmarks and more than 95% accuracy was obtained. The aim of this study was to evaluate the accuracy of the ASM approach as applied to the automatic location of cephalometric landmarks. Besides the land marking, in the diagnosis orthodontists usually trace out the craniofacial structures to carry out some advanced analysis such as superimposition. Superimposition is an important tool to observe and predict the overview of growth changes and evaluate the effect of treatment by comparing different X-rays. As well as land marking, manual structure tracing and superimposition are time consuming, and subject to human error. Some work tried to use the classic image processing technology to obtain the structure extraction. However, simplex image processing techniques (gradients, active contours) enviably fails for some local configurations where gradients are low or inverted.

Romaniuk [18] proposed a regional approach of type shortest path to combine robustness and low algorithmic cost to solve these problems, but his method could only extract the external cranial contour (dome of the skull limited by the nose and the lowest point of the cranium). Actually, due to the nature of cephalograms, it is difficult to use simplex image processing techniques to exactly extract both the soft tissue edges and the structure outlines of internal skull.

Yue et.al.[19] proposed a method in which craniofacial landmark localization and structure tracing are addressed in an uniform framework. Differing from manual procedure, we start with locating the feature points instead of tracking structures, and then trace the structures by connecting the located points with a set of curves based on prior knowledge. In order to support intelligent superimposition on different structures and other advanced analysis, all craniofacial anatomical structures should be extracted in this study. To meet such requirement and allow users

to define new analytical methods, 262 feature points, including 90 cephalometric landmarks and 172 auxiliary points selected according to the structure knowledge, is located.

Lihong Ma et.al.[20] proposed a novel approach to detect landmarks automatically, they use diffusion in different scales in layered segmentation, making uses of region homogeneity and edge saltation , so that the landmark position can be perfectly reflected on edge maps, i.e. skeleton, of layered diffusion. Secondly, key landmarks were determined by binarization pixel cliques which were formed by Euclidean distance map(EDM) on dilated skeleton.30 cases were checked , in which 9 angle parameter and 4 of 5 distance parameter were correctly calculated, the predicted profile were similar to the actual contours except one point that happen due to the intrinsic difficulty in labium prediction .

Kafieh et.al.[21] Proposed a procedure using ASM to automatically detect landmarks in the image, basically they used linear vector quantization (LVQ) to classify images according to the features extracted from each images. The feature is generated by detecting three landmarks automatically. After training the LVQ network, each new image can be judged so the possible coordinate of the land mark are estimated. Then a modified ASM with multi resolution approach is applied and a principal component analysis is incorporated to analyze each template and the mean shape is calculated. Then by performing a local search each model point is shifted to the best location, next a sub image matching procedure based on the cross co-relation is performed to pin point exactly the land mark location, their result shows 24% of the 16 landmarks are within 1 mm of correct coordinates , 61% within 2 mm, and 93% within 5 mm.

I.EI-Fehgi et.al [22] proposed a method to perform the cephalometric analysis of X-ray by image registration approach based on least square approximation. This paper presents a new approach to cephalometric x-ray landmark localization. This approach is based on extracting features from face profile of the lateral skull x-ray. The extracted side profile is then segmented based on knowledge of known local facial structures. Once the face profile is properly segmented, nine perceptually significant landmarks (fiducial points) on the face are registered based on a local maximum curvature computation. These fiducial points are used for point-by-point correspondence to find the transformation coefficients of the control points between two images of a scene. Next, they used the obtained coefficients to find the locations of other landmarks on the target image by mapping the landmarks of the reference image. The algorithm was tested on more than 80 x-ray images to locate 20 landmarks. They mentioned that it was possible to locate

the landmarks by their proposed approach, with accuracy of ±2mm on more 90% of the landmarks.

CHAPTER: IV

AUTOMATED 2-D CEPHALOMETRIC ANALYSIS ON X-RAY IMAGES BY THE EDGE DETECTION APPROACH

4.1: **PROPOSED APPROACH**

we propose a customised approach based on the edge detection technique. The procedure described below can be divided into some steps:

- By knowing the anatomical location of the structure the location of our interest are located by distinctive template matching procedure.
- When the region of our interest are detected a novel filtering approach are applied for de noising the image.
- Then on the filtered image edge detection methodology is performed. For edge detection an inimitable procedure are applied by the modification of original canny edge detection algorithm.
- Morphological opening i.e. repeated dilation and erosion are applied for joining the broken edges.
- Next a local edge linking algorithm is developed for linking the broken edges in the edge maps.
- After that by tracking our desired edge from the edge maps and then plotting it into the whole image the structure are detected into the whole image.
- As the first step for detecting the region of interest we applied parametric template matching algorithm.

4.2: PARAMETRIC TEMPLATE METHOD AND ITS RELEVANCE TO VIGOROUS MATCHING:

The main problem in the conventional template matching algorithm is raised by the difference in shape, size, translation and rotation of shapes, it's position estimation accuracy is limited to one pixel, there is a huge matching cost for large search image and robustness against object deformation and environmental change. so the conventional method must be improved to **(1)** attain high accuracy at a sub pixel level, **(2)** It should lower the matching cost .**(3)** increase the robustness against object deformation and environmental change. The method describe here is

proposed by K. Tanaka et.al.[23], attains these claims through matching between template space constructed from a set of images and a reference image.

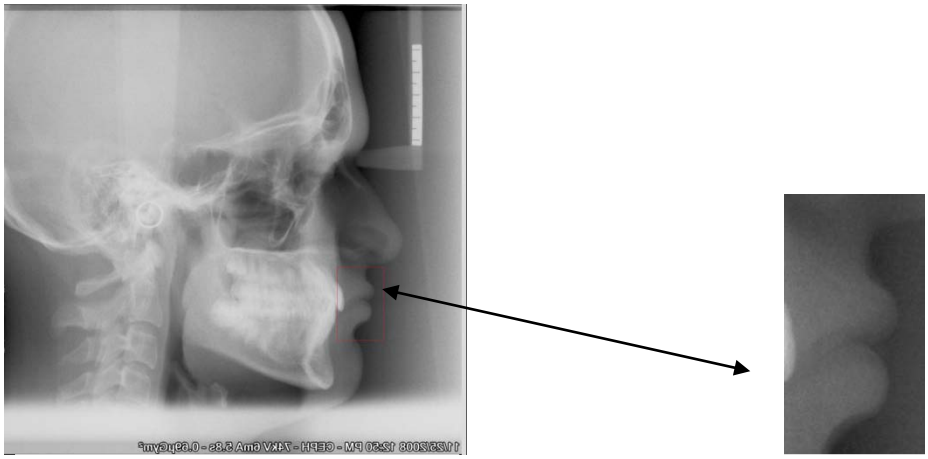


Figure 4.1: Search image and the template image

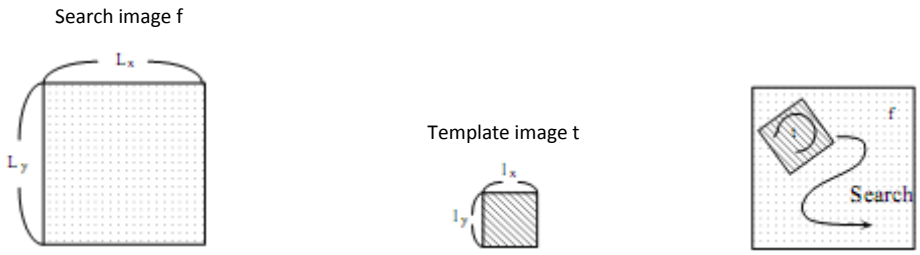


Figure 4.2: Normalised correlation template matching

There is a template which represents each shapes, by moving the template over the image and calculating the cross correlation between them, the highest correlation value is selected as the exact place of landmark. By minimizing the search window for each shape we can minimize the time complexity and also the problem translation or shift in the image since it guarantees that the desired shape will fall inside our searching range. The problem of size difference between the shapes can be partially solved by dividing the complex shape into more than several components. It is easier to locate one complex, large shape if it broken down into several smaller components.

4.2.1 DESIGN OF THE PARAMETRIC TEMPLATE

One way to overcome the problem of rotation & scale, translation and non-geometrical changes such as illumination variations or individual variation between objects in different images is to use multiple templates for each component of the shape but we require a number of templates to represent individual variation of different images is very large and this will increase a large increase in computational cost. a parametric template space is defined as a region of linear template space, which is expanded by a set of base template images $\{t_0, t_1, t_3 \ldots\ldots t_M\}$, where it's linear parameter w_i are all positive or equal to zero and the sum of their linear parameter is equal

to 1. This region is thus forms a simplex having the base templates as it's vertexes in the template space. Any template in the parametric template space is defined by the parameter w_i. We call a template with linear parameters w_i is a parametric template and its base template as a vertex templates. Figure 3 shows the base templates $t_0, t_1 \ldots\ldots\ldots\ldots\ldots t_M$ and the parametric template $*_{t_w}$ with linear parameter w_i . Figure 4 shows a simplex representing the parametric space and its vertex template.

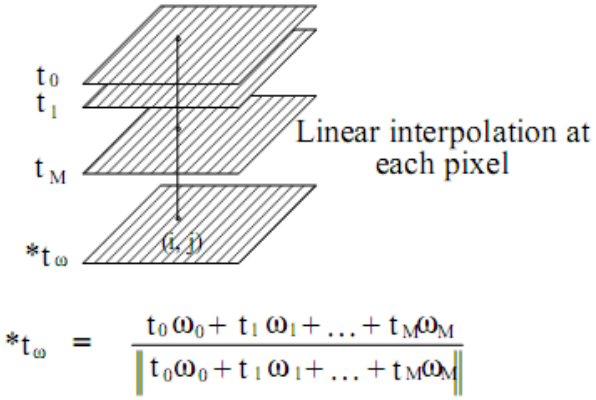


Figure 4.3: Construction of parametric template

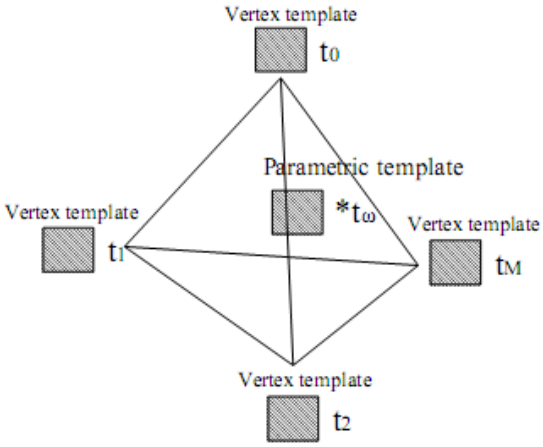


Fig 4. 4: A parametric template space and a parametric template t_w expanded by vertex template t_0 … … … t_M

One solution is to combine a set of template $\{t_0, t_1, t_3 \ldots\ldots t_k\}$ in one template and match in the reference image. A parametric template P can be constructed by using linear combination of all the templates :

$$P = \sum_{i=0}^{k} \propto_i t_i \qquad (1)$$

where t_i is the rotated/scaled version of the template, i varies from 1 to k. k is the no. of shapes represented in one template.

The normalised correlation value ρ between two images $\boldsymbol{g}$ *and* $\boldsymbol{t}$ can be defined by

$$\rho = \frac{\sum_{(k,s)\in TW}(\Delta t)\times(\Delta g)}{\sqrt{\sum_{(k,s)\in TW}(\Delta t)^2}\sqrt{\sum_{(k,s)\in TW}(\Delta g)^2}} \qquad (2)$$

$$\Delta t = t[k,s] - \bar{t},$$

$$\Delta g = g[k,s] - \bar{g},$$

$$TW = \{(k,s) | 1 \le k \le l_x, 1 \le s \le l_y\}$$

where $t[k,s]$ and $g[k,s]$ are the brightness of the images at a location (k, s), and g is the mean value of the image g as defined by :

$$\bar{g} = \frac{\sum_{(k,s)\in TW} g[k,s]}{\sum_{(k,s)\in TW}} \quad (3)$$

Let $\hat{g}$ and $(\hat{g},\hat{t})$ denote the normalization of an image g and the correlation between two image $\hat{g}$ and $\hat{t}$ be:

$$\hat{g}[k,s] = \frac{\Delta g}{\sqrt{\sum_{(k,s)\in TW} (\Delta g)^2}} \quad (4)$$

$$(\hat{g},\hat{t}) = \sum_{(k,s)\in TW} \hat{g}[k,s] \times \hat{t}[k,s]$$

then the normalized correlation ρ can be written as.

$$\rho = (\hat{g},\hat{t}) \quad (5)$$

A normalized image $\hat{g}$ satisfies the following normalised conditions:

$$\sum_{(k,s)\in TW} \hat{g}[k,s] = 0.0$$

$$\sum_{(k,s)\in TW} \hat{g}[k,s] \times \hat{g}[k,s] = 1.0 \quad (6)$$

Now the time come s to change the template matching between the two templates to a template space and a template. If we suppose $t_{0,\ldots\ldots\ldots\ldots} t_m$ are M+1 template images with size $l_x \times l_y$, the linear combination of the normalised templates $\widehat{t_0} \ldots\ldots\ldots \widehat{t_M}$ with parameters w_i (i=0,1.....M) is

$$\sum_{i=0}^{M} w_i \widehat{t_i} \qquad (0.0 \leq w_i \leq 1.0, \sum w_i = 1.0) \quad (7)$$

This linear combination of the template is not normalised , for normalised it should be divided by its norm.

$$*^t{}_w = \frac{\sum_{i=0}^{M} w_i \widehat{t_i}}{\left\|\sum_{i=0}^{M} w_i \widehat{t_i}\right\|} \quad (8)$$

satisfies the normalisation condition (9) and (10) . We call this template $*^t{}_w$ with parameter w_i $(i = 0,1,\ldots\ldots\ldots\ldots,M)$ a parametric template. Fig 3. shows it's construction, and fig 4 illustrate the parametric template space. A normalized correlation between any reference image g and the parametric template $*^t{}_w$ becomes.

$$(\hat{g},\ *^t{}_w)$$

As a conventional rule for finding out the location of the template in the whole image ,we look for maximum correlation with the template and the image, so now consider the problem of maximising the normalised correlation (13) by varying the parameters w_i,

$$(\hat{g},\ *^{t}{}_{w}) \rightarrow \max(\sum_{i=0}^{M} w_i = 1)$$

As this the maximization problem with a constraint and can be found by Lagrange multiplier method. $\overrightarrow{w}$ can be found by :

$$\overrightarrow{W} = \frac{\boldsymbol{H^{-1}}\ \vec{\boldsymbol{G}}}{\vec{\boldsymbol{n}},\ \boldsymbol{H^{-1}}\ \vec{G}}$$

where $\vec{w}, \boldsymbol{H}, \vec{G}\ and\ \vec{n}\ are$

$$\vec{w} = \begin{bmatrix} w_0 \\ \vdots \\ \vdots \\ \vdots \\ w_n \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} \widehat{\boldsymbol{t_0}}, \widehat{\boldsymbol{t_0}} & \cdots & (\widehat{\boldsymbol{t_0}}, \widehat{\boldsymbol{t_n}}) \\ \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{t_n}}, \widehat{\boldsymbol{t_0}} & \cdots & (\widehat{\boldsymbol{t_n}}, \widehat{\boldsymbol{t_n}}) \end{bmatrix}, \qquad \vec{G} = \begin{bmatrix} (\widehat{\boldsymbol{g}}, \widehat{\boldsymbol{t_0}}) \\ \vdots \\ \vdots \\ \vdots \\ (\widehat{\boldsymbol{g}}, \widehat{\boldsymbol{t_n}}) \end{bmatrix}, \ n = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

We can bind the feature value, e.g. a shift value , scaling value of the sub image in terms of following equation.

$$v = \sum_{i=0}^{M} w_i v_i$$

This optimal matching solution between a parametric template and a reference image gives us a fast parametric template matching algorithm:

So the above described algorithm can be ornamented as follows;

1. From a initial template space generate a set of vertex templates $t_{0,\ldots\ldots\ldots\ldots\ldots} t_M$.
2. Calculate correlation matrix **H** among the vertex template.
3. Calculate the correlation vector $\vec{G}$ which consists of correlation values between a reference image g and the vertex template.
4. Determine $\vec{w}$ and v according to the equation (15) and (20).

In this approach the optimal parameters $\vec{w}$, which maximize the normalized correlation value (13), and the feature value v are directly obtained from the normalised correlation vector $\vec{G}$ and the correlation matrix H without generating the parametric template images at each parameter. So this algorithm enables a fast matching between the parametric template and the reference template. Figure below shows the templates with five degrees of rotation: +2.5,-2.5,+5,-5,0 (there will be five copies for each template). Construct a weighted linear sum of the above five templates as a new template space. Then find the 2D cross correlation of the selected window of test image with new template space. Figure below shows the construction of template space.

To reduce processing time and eliminate risk of false alarms, pattern matching is applied only in the zones determined by a prior knowledge of human skull. In general for detecting the regions

separately at first the fixed tripod rod which is common in every image and is located in the first quadrant of the image is detected. Assuming the location of the rod as an entry region of the image, and based on the location of the rod in the test image the nose region are detected, and based on the nose location the lip region are detected and by proceeding in this manner all the remaining region of interest are detected. Figure (2) shows the results of desired region detection.

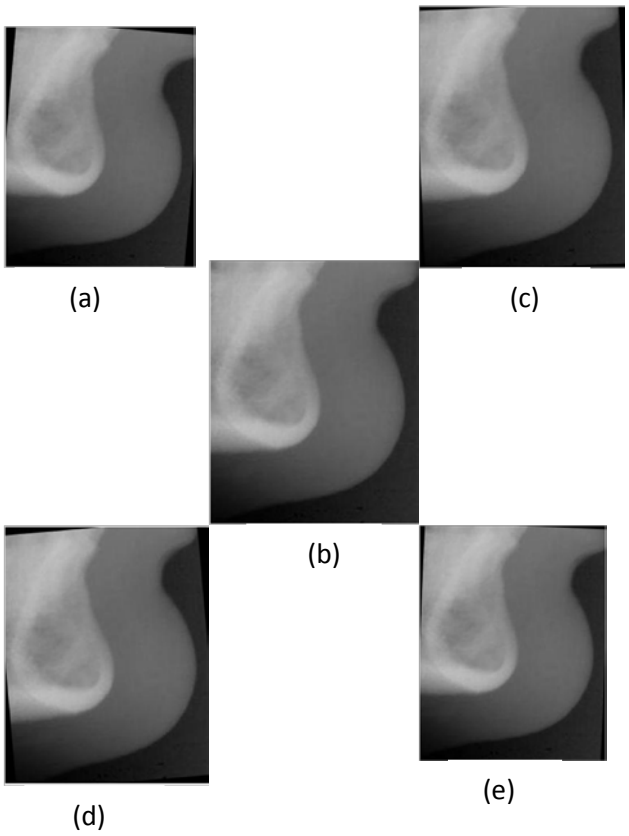


Fig 4.5 (a) construction of template space

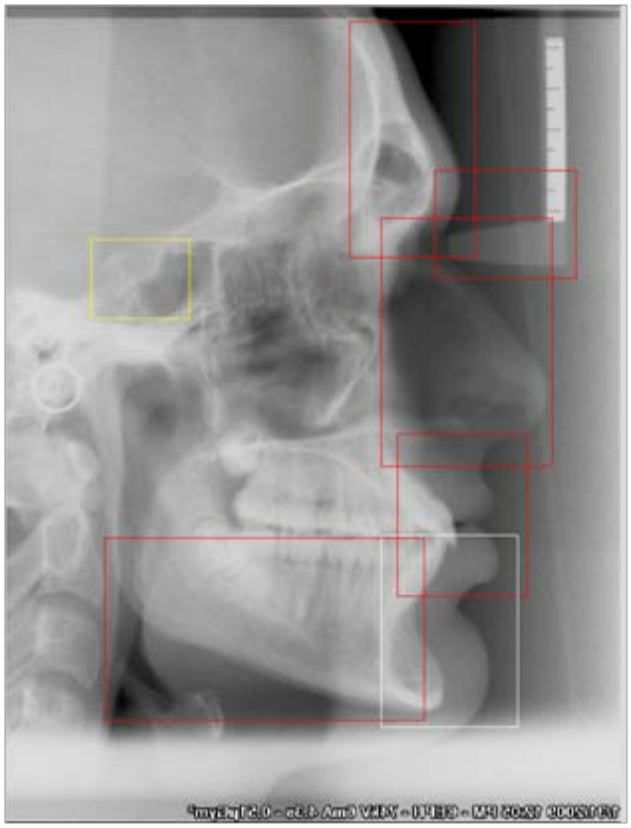


Fig 4.5: (b) Results of region detection by template matching approach

4.3 ADAPTIVE NONLOCAL FILTERING:

Many early vision systems employ some type of filtering in order to reduce noise and/or enhance contrast in regions that correspond to borders between different objects within an image. Filtering with a kernel that is symmetric around the centre pixel results in averaging of edge values therefore it also results in blurring of the edges. In order to alleviate this problem Nitzberg and Shiota proposed an offset term which pushes the centre of the kernel from the point to be filtered. So the purpose of generating this type of offset vector field is to displace the filter away from the border areas. The main purpose of this type of offset vector field is to displace the filters away from the border areas to preserve the edge information at the border area. B Fischel et al. propose this displacement in the direction normal to the boundary of a region, as this is the direction with no component of edge. In order to compute this type of vector field, three issues must be addressed, each of which depends on an estimation of local edge, if one exists.

At first the normal vector are determined since the gradient is normal to the level sets of image, using the gradient direction as the estimation of the normal is a reasonable approach. Then a sign assigned to the normal vector i.e. determination should be made as to whether the displacement in the increasing or decreasing gradient, i.e. the decision reflects, to which region the point in

question has been assigned- the region at the top of the gradient or at the region at the bottom of the gradient. The final aim is to displace the filter away from the presumed edge location. Once the normal vector and its sign have been fixed, the magnitude of the displacement must be determined.

An offset vector field is estimated by

$$O(z) = m(z)d(z)\frac{O(z)}{|O(z)|} \quad , z = [x\ y]^T \tag{1}$$

$O(z)$ In (1) is the offset orientation, and refers to the choice of the normal vector. Offset direction $d(z)$ is a binary one (1 or -1) with the choice of the sign of the normal vector. The offset direction term determines whether offset vector is in the orientation direction or in the opposite direction. Finally the offset magnitude $m(z)$ denotes the length of the offset vector.

4.3.1 OFFSET ORIENTATION

The orientation of the offset vector field at each point in an image represents the estimated orientation of the local edge, if one exists. Generally we desire offset orientation $O(z)$ to be orthogonal to the local edge, and hence normal to the boundary. To accomplish this in a manner insensitive to noise is to use gradient of the smoothed image. Denoting the smoothed image by I, the offset orientation at the point is then given by

$$O(z) = \nabla I(z) \tag{2}$$

4.3.2 OFFSET DIRECTION

Once the offset orientation has been fixed by (2), the direction of the offset field is computed; with it the sign of the normal vector is computed. So the choice of direction is thus a binary one and can be chosen as a local segmentation decision, saying whether the current point has been assigned to the region at the top of the local intensity gradient or to the region at the bottom. This decision is made per pixel basis.

$d(z)$ is the offset direction, it basically determines which side of the centre of the local edge the current point is on, it is computed by two criteria. The first is that the gradient will contribute to the choice of direction. The second is that gradients of the proper orientation on one side of a point with respect to the offset orientation should contribute towards a displacement in the opposite direction. Combining these two constraints the following equation are derived for the calculation of offset direction.

$$d(z) = -sgn\left(\int_W (O(z).\nabla I(z + z')(O(z).z')dz'\right) \quad , \tag{3}$$

Where sgn is the sign function and z' is the relative coordinates relative to the filter centre. If $d(z)$ is positive, the displacement direction is in the orientation direction, otherwise it is in the opposite direction. The first term in (3) reflects the given criterion stated above, it weights gradients in the offset direction more strongly than those at the other orientation. This term is important as it discounts noise and corner induced gradients.

The second term indicates the second constraints; it is very similar to computing the first moment. It measure the contribution of a point based on its distance in the direction normal to the local edge. Points with an appropriate gradient (as computed via the first term) that are distant from the central point indicate the presence of an edge in that direction and hence push the offset in the opposite direction. If d (z) is positive the displacement direction is in the orientation direction, otherwise it is the opposite direction.

4.3.3 OFFSET MAGNITUDE

Once the offset direction and orientation is fixed, then we determine the offset magnitude i.e. how far the displacement should be in the selected direction. Finding out the offset magnitude is a critical decision, as the displacement will be large in broad edges and small in the area which contains small-scale image structure. To satisfy both this constraints we perform a 1-D search mechanism which allow offset magnitude to grow based on the outside of window W. Basically what we does, we search in the offset direction for a zero-crossing of the vector field in that direction (i.e., until the dot product of the offset in the central point with any point in the offset direction is non-positive)

$m(z)$ Refers to offset magnitude, a one dimensional search is performed in the offset direction to find zero crossing of the vector field in that direction (i.e. until the dot product of the offset at the central point with a point in the offset direction is no positive). It indicates that the vector field is either vanished, signifying the interior of a region, or has changed orientation by 90 degree, that possibly indicates the presence of the far edge in the region. So dot product generates a barrier which prevents the offset vector from extending across additional edges in the offset direction.

So offset calculation is performed by the two step process. First by creating a offset field (2)-(3).via

$$v_i(z) = d(z)O(z). \quad (4)$$

Next, we search in the offset direction for the first point z' such that the dot product of the initial offset at z' with the initial offset at the central point z is non positive.

$$m(z) = min\alpha: v_i(z + \alpha v_i(z)).v_i(z) \leq 0 \quad (5)$$

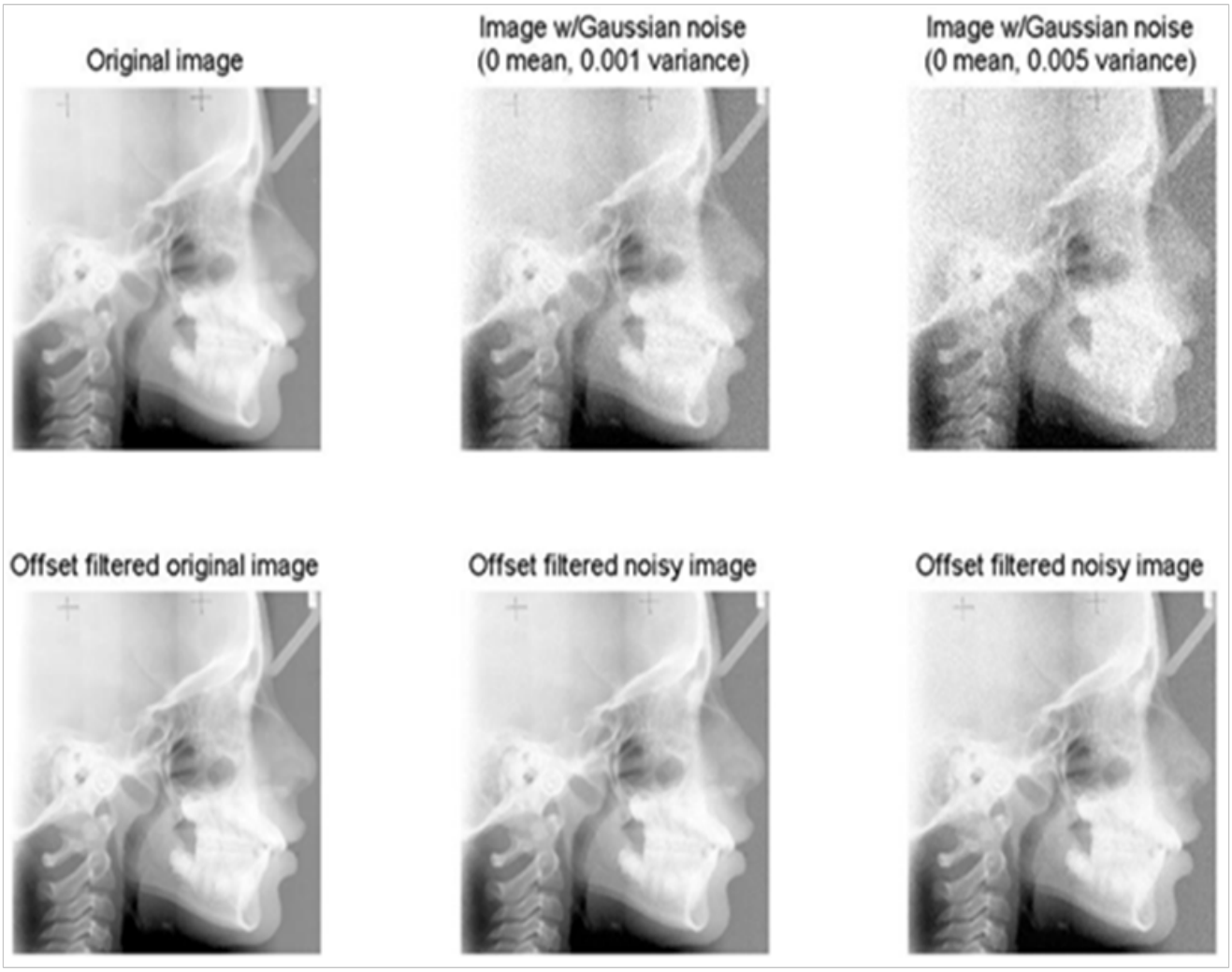
Finally, we form the final vector field using $m(z)$ as the magnitude of the initial vector field.

$$v(z) = m(z)\frac{v_i(z)}{|v_i(z)|} \quad (6)$$

It is really important to note that the adaptive offset magnitude adds a different feelings of scale into the offset filtering than is usually used in contemporary applications of diffusion or scale space architectures. The diffusion approaches grew out of linear filtering techniques such as those Burt [24], Witkin [25] and Marr [26]. In these approaches, the scale of the features is defined by the size of the kernel required to detect it. In the anisotropic extension of diffusion approach, scale is associated with integration time modulated by local gradient magnitude, and by the extension across which intensity value diffuse to arrive at a given location. Region of high gradient bottle up the amount of diffusion, and so are associated with smaller scale than smoother image areas. The integration of anisotropic diffusion is therefore results in intensity values near edges being replaced with smoothed versions of internal intensity values from the direction away from local edge.

In B. Fischl's approach the relation between the scale and distance made explicit via the magnitude of the displacement vector at a given location. Large scale (i.e. more blurred) edges result in longer displacement vector but no change in vector size where small scale image feature result in small displacement vector as it constraints the length of the displacement vectors, so it preserves the edges of our interest. The smoothing associated with smoothing can be accomplished any of the variety of standard fixed size filter which are applied none locally at the offset location.

Figure below (4.6) shows the effect of offset filtering on the image.



(a) (b) (c)

Figure 4.6: (a) original image and the result of filtering. (b), (c) original image with added noise and the result of filtering.

4.4. MODIFICATION OF CANNY'S EDGE DETECTION ALGORITHM

The edges of the image are detected by canny edge detector. In this method generally, the image is first smoothed by Gaussian filter to estimate the noise in the image .Then the spatial gradient calculation is performed by the Gaussian kernel so that gradient magnitude, Gx for horizontal direction and Gy for vertical direction for each pixel can be measured. The edge direction of every pixel can be measured by$\theta = \arctan\left(\frac{Gx}{Gy}\right)$. Next we select a suitable pair of high threshold value (HTV) and sigma. The low threshold value (LTV) is automatically selected as 0.4×HTV. However, it is very difficult to select a suitable pair of HTV and sigma for detecting soft tissue

profile as well as hard tissue profile in a set of cephalometric images due to the local intensity variability and low contrast of the hard tissue profile against the soft tissue.

Next we perform the non maximum suppression in the edge direction and suppress any pixel whose gradient magnitude is not at maximum, this non maximum suppression gives a thin edge in the later processed image. Then we check each pixel returned by the non-maximal suppression operation against the LTV and then against HTV. Any point whose magnitude is greater than HTV are chosen as edge point, any pixel whose value is less than LTV are choose as non edge pixel and the pixel with value greater than LTV but less than HTV are selected as candidate points, they are considered as non edge unless there is a direct or indirect path from this pixel to pixel with a gradient above the high threshold.

The use of HTV and the LTV picks up more meaningful edges compared to other edge detector which uses only single threshold value (sobel, prewitt, Roberts). The hysteresis thresholding with a smaller HTV and/or smaller LTV can certainly pick up more edges, but may also collect a large portion of noisy edges. On the other hand, a larger HTV and/or a larger LTV can certainly pick up a high quality of edges at the expense of losing of some true edges. In Canny's method choosing the HTV and LTV is based on the analysis of the entire image, so as usual it can result in the loss of weak edges.

To overcome or alleviate this problem, a more sensitive approach is to automatically select the HTV and LTV based on a local neighbourhood of a pixel rather than from the whole image. For the detection of craniofacial structure in cephalometric images, the selection of both LTV and HTV can be made empirically by simple experiments. By inspecting the output of edge detection using different threshold values, a suitable pair of LTV and HTV can be determined. However it is much more difficult to select a suitable pair of LTV and HTV for detecting hard tissues due to the local intensity variability and low contrast of the hard tissues.

That was the basic algorithm of canny edge detector [29] and this original canny edge detector failed to detect some of desired edge so we applied some modification of canny edge detection technique proposed by Samuel H.Chang et al.[28] .

The modified canny edge detector is developed to determine the HTV and LTV based on the local connection of any selected edge point. The algorithm is stated below:

Step 1: The edge points detected by Canny, the location of the candidate points, the magnitude of the entire pixel are selected.

Step 2: The Eigen value map of the image is calculated by this way: a) A ROI is selected which cover the structure of our interest b) the Eigen value of a pixel is the minimum Eigen value of the matrix of the neighbourhood centred at the pixel.

Step 3: A threshold value of the Eigen value map is selected, in our case we select it as the ' (maximum + minimum)/2 'of the Eigen value matrix.

Step 4: For any pixel with its corresponding Eigen value greater than the threshold value, if it's neighbourhood contains a candidate point and a edge point cross the pixel i.e. the candidate point, centre pixel for which the neighbourhood is formed and edge point will lie in a straight line, then the gradient of the candidate point is selected as local dynamic HTV.

Step 5: Select new edge points in the locality using the local dynamic HTV and the global LTV.

Figure 11 shows such an application of the algorithm described above in which modified canny detector detects some desired edge which cannot be possible by normal canny edge detector.

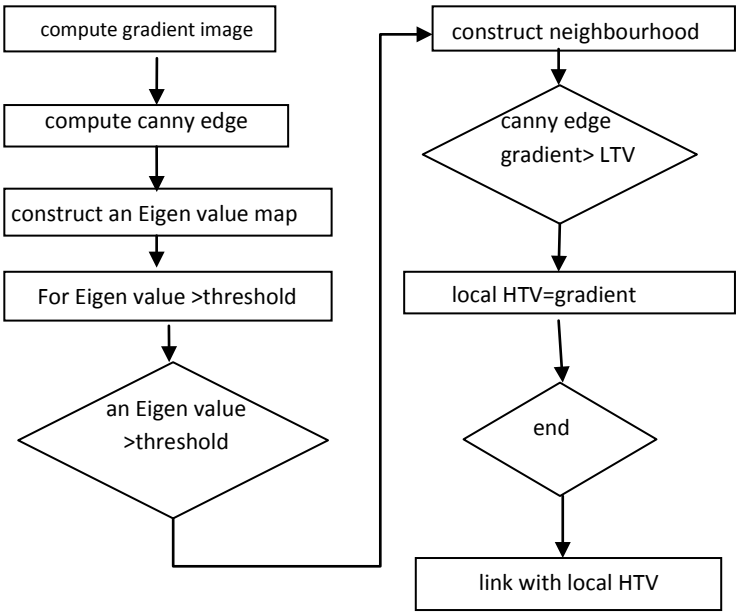


Figure 4.7 : Flow diagram of the algorithm

The above flow chart demonstrates the flow of the algorithm, figure below shows the result of proposed edge detection methodology. It demonstrates how the modified Canny's edge detection algorithm are able to detect the small scale edges which is not possible to detect by the original Canny's edge detection algorithm.

(a) (b) (c)

Fig 4.8: (a) Shows the original image. (b)the edge detected by canny edge detector with HTV =0.4 (c) the portion encircled by red is the edge detected by the algorithm

4.5 EDGE LINKING

Edge detection is an important step in image processing and plays a role in digital image segmentation, stereo matching, and object recognition, medical image processing and automatic navigation. One of the primary problems with digital edge detection is that when the background and foreground are of similar colours the edge detector may pick up only a fraction the edge pixel that form the edges. Properties such as alignment and symmetry allow a human to easily connect the "missing edge points". Indeed, human perception is the best-known method for linking the edges. Even without knowledge of what the actual object is, it is easy for a human observer to link the gappy edges.

One novel edge linking technique is proposed by Jamy Li et.al.[27] that attempt to simulate approaches used by humans. At first pre-processing the input edge image in several steps. The pre-processing step involves performing non-maximal suppression and thinning (through local processing technique) to the edge image so that it became thinned or skeleton image (or nearly so). This renders binary image. When this is done, point classification can be performed, which classifies all points as end points, edge points or intersection points.

4.5.1 POINT CLASSIFICATION:

This method requires point classification as a initial step. Its purpose is to classify each pixel as an initial step. Its purpose is to classify each pixel as one of the three types: end points, edge points and intersection points. This is done by setting up a look up table for finding an end points and intersection points among the edge points**.** We create all possible 3×3 neighbourhood of the total edge image and check whether the centre pixel within this 3×3 neighbourhood is end point or a junction point. This is done by creating a constraint that to be checked for end point or

intersection point , the centre pixel must set and the number of transitions/crossing between 0 and 1 as one traverse the perimeter of the 3×3 region must be 6 or 8 for intersection point and 2 for end point.

4.5.2 CREATING A LOOKUP TABLE

Most easily we can implement certain binary operations through lookup tables. Basically a lookup table is a column vector in which each element represents the value to return for one possible combination of pixels in a neighbourhood. The following figure represents these types of neighbourhoods. Each neighbourhood pixel is represented by an x and the centre pixel is one with underlined.

x	x
x	x

2×2 neighbourhood

x	x	x
x	**x**	x
x	x	x

3×3 neighbourhood

In the case of *2×2* neighbourhood, there are 16 possible permutations of the pixels in the neighbourhood. Therefore, the lookup table for this operation is a 16 element vector. For a *3×3* neighbourhood, there are 512 permutations, so the lookup table is a 512-element vector. Lookup table larger than 3-by-3 is not practical. The vector consists of the output values for all possible 2-by 2-by-2 or 3-by-3 neighbourhoods. By performing a neighbourhood operation on a binary image by producing a matrix of indices into the lookup table, and then replacing the indices with the actual values in the lookup table the specific algorithm used depends on whether you use 2-by-2 or 3-by-3 neighbourhoods. For 2-by-2 neighbourhoods, length (lookup table) is 16. There are four pixels in each neighbourhood, and two possible states for each pixel, so the total number of permutations is $2^4 = 16$.

To produce the matrix of indices, the binary image BW is convolves with this matrix.

8 2

4 1

The resulting convolution contains integer values in the range [0, 15]. The central part of the convolution are used, It is of the same size as BW, and adds 1 to each value to shift the range to [1, 16]. It then constructs A by replacing the values in the cells of the index matrix with the values in lookup table that the indices point to.

For 3-by-3 neighbourhoods, length (lookup table) is 512. There are nine pixels in each neighbourhood, and two possible states for each pixel, so the total number of permutations is 2^9= 512

To produce the matrix of indices, the binary image BW is convolved with this matrix.

256	32	4
128	16	2
64	8	1

The resulting convolution contains integer values in the range [0,511]. The central part of the convolution is used, it is of the same size as BW, and adds 1 to each value to shift the range to [1,512]. It then constructs A by replacing the values in the cells of the index matrix with the values in lookup table that the indices point to.

After the step is converged we perform edge linking. For this edge linking a second level hysteresis is used, and it is a noble development on regular hysteresis thresholding. In the regular hysteresis there exists some difficulty in identifying the edges; basically thresholds are used for eliminating the weak edges so gaps are also created. General hysteresis therefore demonstrates a trade-off between eliminating noise and fragmenting edges.

The basic idea is to use two edge images that have undergone hysteresis: a high image (obtained by applying a HTV) and a low image (obtained by applying a LTV). Naturally the high image contains only strong fragments of edges; whereas low image contains both the strong and weak edges.

The main idea is to use the high image as guidance for promoting edges from the low image i.e. expressing them in the output image, so output image will consist of edges from high images and those edges which are promoted from the low image. The criterion for promotion of the edges from the low image is connectivity, whether the two strong edges in the high image is joined by a weak edge in the low image, if it happens then that edge from the low image will be promoted in the final result. The algorithm is as follows:

We subtract the low image edge points from the high image, it forms the difference image. We find the end points in the high image. For each end point (end points are detected according to module 4.4.2) in the high image we search the neighbourhood of the end points in the difference image if any edge pixel are found then that neighbouring edge are traced to see whether it reaches to another end point in the high image. If connection is found, the line in the difference image is selected.

For making this strategy to be effective there must be a significant difference in the thresholds used for the high and low images. However selecting the HTV and LTV is a very critical job. In addition, the linking of this gap of edges also creates some unwanted branches of the edge; this can be eliminated later stage by tracking algorithm.

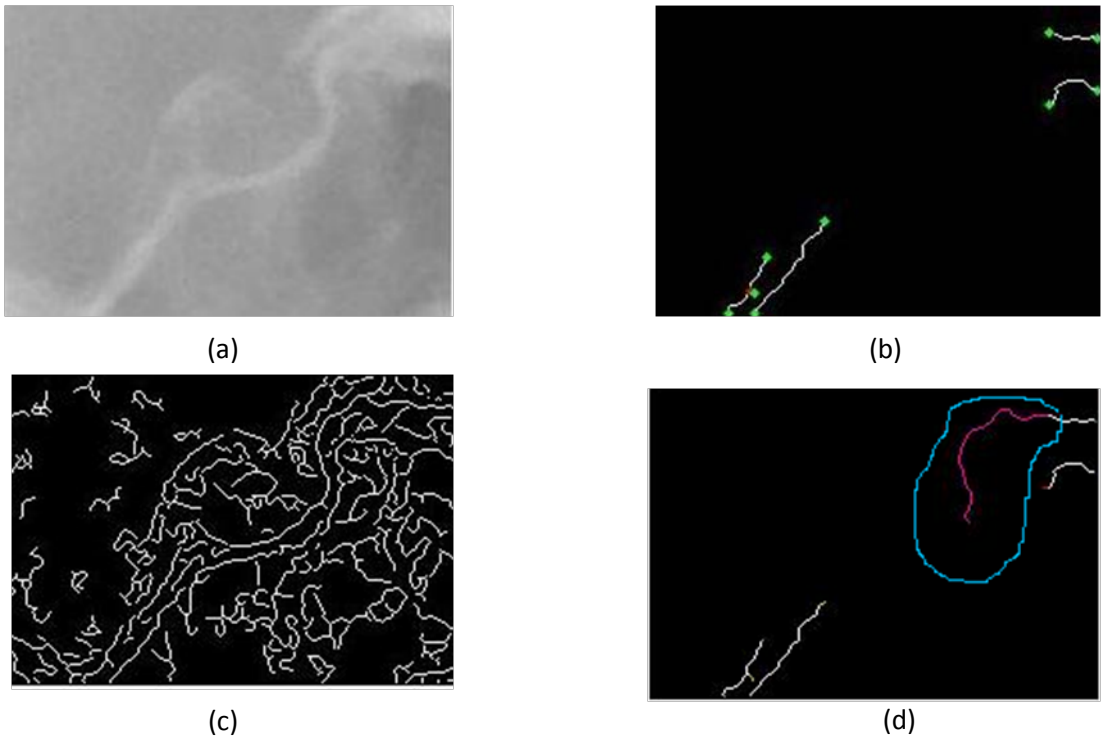


Figure 4.9 : (a) The original image (b)The edge detected by high threshold value of 0.6 & the detection of the end points in green (c) The edge image with high threshold. (d) The region encircled by sky blue are the edge obtained by the algorithm

Joining end points that are close together is meant to compensate for the consequent gaps in edge lines.

4.6 EDGE TRACKING MODULE: After the result performed by edge detector, the edges of the image are obtained, which is shown as a white pixel and the non edge points are shown as a black pixels. The gray scale value of the non edge pixel is 0. The gray scale value of the edge pixel is 1, this kind of image is known as binary image containing only 0 & 1. Now on the binary image the edge tracking algorithm is performed and put forward a bidirectional contour tracking algorithm. The edge tracking algorithm can be described into some steps that are described below.

The introduction of this algorithm begins with the choosing of an edge point as the starting point at the pull of edges. Suppose the binary image matrix is M, the coordinate of the first edge point is (x_0, y_0), and the next edge point must be located in its eight neighbourhoods of it. The coordinate coding of the eight neighbourhoods is shown in figure (2a). Figure (2b) denotes

the location coding of the eight neighbourhood pixel. In our proposed algorithm we have applied a bidirectional mode contour tracing approach proceeds by tracking the edge in both directions one after another.

(x-1,y+1)	(x,y+1)	(x+1,y+1)
(x-1,y)	(x, y)	(x+1,y)
(x-1,y-1)	(x,y-1)	(x+1,y-1)

Coordinate denotation

1	8	7
2	(x, y)	6
3	4	5

Coding Denotation

III. Edge Tracking: Ascertain the initial point is in M, (we perform a raster scan in the image until the first white pixel is not found, generally we start scanning from 2^{nd} row & 2^{nd} column to perform the neighborhood operation) the confirmation of **edgelist{i}** is special, where **edgelist{i}** is a two dimensional matrix holding the coordinates of the edge pixels. From figure (2) we can determine the next edge point in **edgelist{i}** must be located in the neighborhood i.e. at the location 2,4,6,8 or 1,3,5,7 region of our selected first edge pixel. So a neighborhood is created centering this pixel, and checks whether there is any other edge pixel (value =1) in its neighborhood, if the edge pixel exits, pick the location of the edge point.(detail is described below)

The edge tracking algorithm can be described into some steps mentioned below.

1. As we track the edge on the binary image, so binary image are used as the input to the tracking algorithm.
2. The isolated pixels (individual 1's that are surrounded by 0's),are removed from the image, such as the centre pixel in the location.

0	0	0
0	1	0
0	0	0

Matrix representation of isolated pixel

3. Thinning operation are performed on the image by maintaining the following algorithm.
   (a) Divide the image into two distinct subfields in a checkerboard pattern.
   (b) In the first sub iteration, delete pixel p from the first subfield if and only if the conditions G1, G2, and G3 are all satisfied.

(c) In the second sub iteration, delete pixel p from the second subfield if and only if the conditions G1, G2, and G3' are all satisfied.

Condition G1: $\mathbf{X_H(p) = 1}$ where $\boldsymbol{X_H(p) = \sum_{i=1}^{4} b_i}$; $\boldsymbol{b_i = 1\ if\ x_{2i-1} = 0\ and\ (x_{2i} = 1 or x_{2i+1} = 1)}$

$x_1, x_2, \ldots\ldots\ldots\ldots x_8$ are the values of the eight neighbours of p, starting with east neighbour and numbered in counter clockwise order.

Condition G2: $2 \leq \min[n_1(p), n_2(p)] \leq 3$ where $n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$,

$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$

Condition G3 : $(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$

Condition G4 : $(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$

The two sub iterations together make up one iteration of the thinning algorithm. When the user specifies an infinite number of iterations (n=Infinite), the iterations are repeated until the image stops changing.

4. Perform a raster scan in the image until the edge pixel is not found. Let's the coordinate of the first edge pixel is (rstart, cstart). Make M (rstart,cstart) = - curvno (Where curvno. is a variable initialized to zero)

5. A neighborhood is created centering this pixel, and checks whether there is any other white pixel (value =1) in its neighborhood, if the white pixel exits pick up the edge point as the part of edge and marked it as a collected point (to avoid repetition) and go to step 6. If it does not exist go to step 4.

6. A neighborhood is created by centering the pixel obtained from previous step, (say (p, q)), and checks whether there is any other edge pixel (say (s, r) whose value =1) in its neighborhood, if the white pixel exits pick this edge point as the part of the current edge and marked it as a collected edge point, (i.e. make M (s, r) = -curvno) if the following condition is satisfied

• If this edge point ((p,q)) itself have less than two neighbor (i.e. two white pixel) connected back to our current edge,(i.e. which is not marked as collected edge point) then repeat this step, (go to step 6) until end of the edge, if it does not exist go to step 7.

7. Now repeat the same step for tracking in the opposite direction from the location (rstart, cstart) (described in pseudo code), i.e. go to step 6.

8. Make curvno = curvno+1, perform the raster scan in the image until the untraced white pixel does not obtained, let the location of the pixel is (rstart, cstart). Go to step 6.

Continue in this manner until all the edge points are not traced.

```
//EDGEIM is the binary image after thinning operation
[Row, Cols] = size (EDGEIM)
//getting the total no. of rows & cols in the image
edgeNo = 0  // A variable used for indexing
for (r=1:1:Rows)  //running for whole image
for (c=COLS:-1:1)
  if EDGEIM (r, c) = =1//checking for edge pixel
  edgepoints = trackedge (r,c,edgeNo);
        if edgeNo==0
           edgeNo=edgeNo+1;
           edgelist{edgeNo} = edgepoints;
       end
    end
end
end
edgeim =-EDGEIM; //Finally negate image to make edge encodings positive now  according to the following algorithm, eliminate  isolated edges and spurs that are below the minimum length.(detailed algorithm are described below)

function edgepoints = trackedge (rstart,cstart,edgeno)
    edgepoints = [rstart, cstart];
    EDGEIM (rstart, cstart) = -edgeNo;
    [status,r,c] = nextpoint (rstart , cstart, edgeNo)
       while status not equals to nopoint
           edgepoints = [edgepoints
                         r        c];
          EDGEIM (r,c) = -edgeNo;
          if status = = endpoint;
            status = nopoint;
         else
            [status , r, c] =nextpoint (r,c,edgeNo)
         end
end
edgepoints = flipud(edgepoints); //flipping of edge list for //bidirectional edge tracking
[status,r,c] = nextpoint (rstart,cstart,edgeNo) //creating neighbourhood while status not equals to nopoint
    edgepoints = [edgepoints
                      r    c  ];
         EDGEIM (r, c) = -edgeNo;
            if status==lastpoint
               status=nopoint;
            else
         [status, r, c] = nextpoint (r, c, edgeNo);
```

```
            end
      end
```
[status , nextr , nextc] = nextpoint (rp , cp, edgeNo)
```
          roff = [-1 0 1 0 -1 -1 1 1];
          coff = [0 1 0 -1 -1 1 1 -1];
            r = rp + roff;
            c = cp + coff;
           ind = find (( r>=1 & r<=Rows) & (c>=1 &c<=Cols ))
               checkflag = 0;
                for i =ind
                   if EDGEIM (r (i),c(i)) = = 1 //if there is any edge  pixel
```
at the neighbourhood of (rp, cp), (say (r(i),c(i))) then return first connected edge point (r(i),c(i)) which itself has less than two neighbours connected back to our current edge

n = neighbours(r(i),c(i));
```
        if sum(n==-edgeNo) < 2
           nextr = r(i);
           nextc = c(i);
           status = thereIsAPoint;
           return;
        else
           checkFlag = 1;
           rememberr = r(i);  //see below
           rememberc = c(i);
        end
            end
            end
          if checkFlag
             nextr = rememberr;
             nextc = rememberc;
             status = thereIsAPoint;
             return;
          end
             nextr = 0;
             nextc = 0;
             status = noPoint;
```
//rememberr, rememberc, thereIsApoint, noPoint all are variables where thereIsApoint initialized to 1& noPoint initialized to 0.

function n = neighbours(rp, cp)
```
      roff = [-1  0  1  1  1  0 -1 -1];
      coff = [-1 -1 -1  0  1  1  1  0];
       r = rp+roff; //generating 8 neighbour
       c = cp+coff;
       ind = find((r>=1 & r<=ROWS) & (c>=1 & c<=COLS)); //finding no.
```
of occurrence of (r, c) in full image, which equals to 8
```
       n = zeros(1,8);  //creating matrix (1 8), where each cell are initiated to zero.
       for i = ind
        n(i) = EDGEIM(r(i),c(i));
       end
```

Fig: 4.9: Pseudo code for Edge tracking operation

Then for removing the unwanted edges of short length, we remove them by generating a constrain that if the length of the collected edge is not greater than the certain length (Length of the edge defined by user by knowing the geometrical definition of the location and having the previous knowledge about the structure to be extracted) then remove those edges. The unwanted edge removal approach can be described as follows.

• Each significant edge has two nodes (as considered), the start node and the end node, so number of nodes (say N) = 2 × number of edges, A matrix (N×2) is created, which sequentially contains the entire start node and the end node of each edge.

• Create an adjacency matrix for the nodes (A) and for the edges (B) for keeping the records of connectivity.

• If we sum up the column of the adjacency matrix, we get the number of other edges are connected to an edge and number of other nodes are connected to a node.

• Check every edge to see if any of its ends are connected to just one edge, then we simply merge it with the edge it is connected to.

• Check every edge against the minimum length, suppose length of edge **E1** is less than the minimum length, then find out the number of other edges which are connected to **E1** and the subsequent start node (s1) & end nodes (e1). If number of connection to **E1** is 0 then **E1** is an isolated edge, so remove the edge (removing an edge also includes performing the necessary updates to the edge adjacency & the nodes adjacency matrix). If nodes connected to s1or e1 is 2 (look out from matrix A) then check out the number of edges connected with **E1**, if it is one then this edge definitely will be a loop, so remove the loop.

If it does not happen then find out the length of **E1** and the number of edges connected to **E1**. Suppose there are two (p1&p2) such edges. Now check number of edges connected to p1 (say m) and p2 (say n, where m>n). Now check out the length of E1, p1 & p2. The shorter one (suppose p1) will be selected for removal, but before that find out the start node & end nodes of p1, find out other nodes (say n1,n2) are connected with this start node or end node. Now from matrix A it can be determined that what are the nodes (say node1 & node 2) are connected with n1, n2, if it happens then merge node1 & node2 and update the matrix A.

• Now finally clean up any new isolated edges that might have been created by removing the unwanted edges. Then as a last work, collect all the existing edges.

4.6.1. Merging nodes

If it is to be said to merge two nodes n1&n2, at first the edges (say edg1& edg2) consisting of the nodes n1 & n2 are obtained. Then the start node & end node associated with edg1 & edg2 each are obtained. Now check whether edg1& edg2 are same or not, i.e. whether the entry in the **A** matrix in the $n1^{th}$ row & $n2^{th}$ column is 1or 0. Now check n1 & n2 each, whether any of them are start node or end node. There are four conditions to be considered:

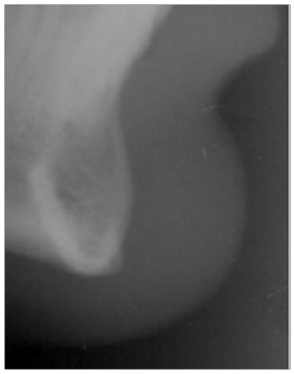
A. n1 of edg1 is end node and n2 of edg2 is start node, then join edg1& edg2 and modify the **A** matrix as all the entries in $e1^{th}$ row & $e1^{th}$ column will be equal to $e2^{th}$ row & $e2^{th}$column respectively. Number of connection to e1 node will be equal to number of connection of e2 node.

B. n1 of edg1 is end node and n2 of edg2 is end node, then flip (i.e. edge composed of pixels collected in a r×2 matrix will be flipped w.r.t central item so that r^{th} entry will be the 1st entry & 1st entry will be r^{th} entry) edg2 and merge edg1 & edg2, modify **A** matrix as all the entries in $e1^{th}$ row & $e1^{th}$column will be equal to $s2^{th}$ row & $s2^{th}$ column respectively. The number of connection to e1 will be equal to s2.

C. n1 of edg1 is start node and n2 of edg2 is start node, then flip edg1 and join with edg2. Update **A** matrix as all the entries in $s1^{th}$ row & $s1^{th}$ column will be equal to $e1^{th}$ row & $e1^{th}$ column respectively and all the entries in $e1^{th}$ row & $e1^{th}$ column will be equal to $e2^{th}$ row & $e2^{th}$ column respectively. Number of connection at s1 of edge1 will be equal to number of connection of e1, and number of connection at e1of edge1 will be equal to number of connection of e2.

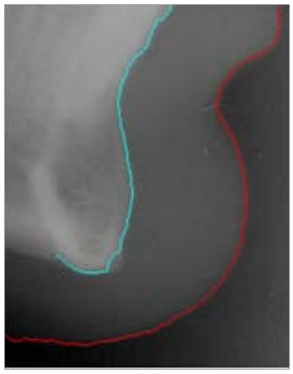
D. n1 of edg1 is start node and n2 of edg2 is end node, then flip edg1& edg2 and join edg1&edg2. Update **A** matrix as all the entries in $s1^{th}$ row & $s1^{th}$ column will be equal to $e1^{th}$ row & $e1^{th}$ column respectively and all the entries in $e1^{th}$ row & $e2^{th}$ column will be equal to $s2^{th}$ row and $s2^{th}$ column respectively. Number of connection to s1 will be equal to number of connection to e1, and number of connection at e1 will be equal to no. of connection to s2 of edg2.

Now correct the edge adjacency matrix to reflect the new arrangement, the edges that are new edg1 is connected, will be all the edges edg1 & edg2 were connected to. The connection counts are recomputed, as the adjacency matrix has been shuffled. Then finally remove edg2, because edg1&edg2 are joined now. After the final edges have been extracted, then the coordinate of the edge points are mapped onto the original image.



(a)

(b)



(c)

(d)

Figure 4.10 : (a) Template extracted from the main image, (b) Result of edge detection on cropped image, (c) edge map on (5 a), (d) Edge map on the full image.

4.7 RESULTS :

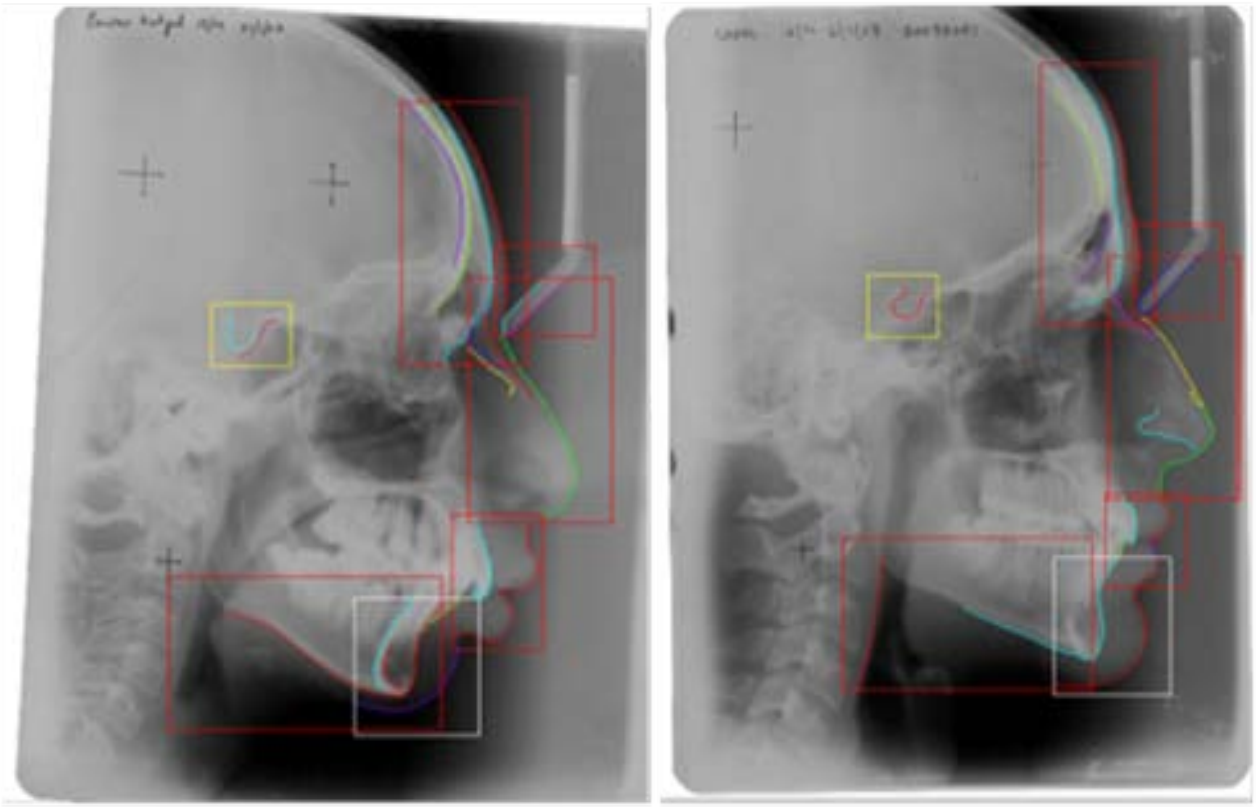
We applied our experiment in two set of images. By the experiment on the dataset1 & dataset2, and considering a limit (suggested by doctor), if the particular structure is detected more than 80% of the required detected length of the structure then that detection is considered as an acceptable detection, the result shows that the hard tissue profile of chin is detected in almost every images and soft tissue profile of chin is detected in 83-85% percent cases. The angle of mandible and its lower border are detected absolutely in 21 images but in rest of 9 cases the structure is partially detected, among those 6 cases is acceptable as detected length crossed above mentioned threshold value, table below illustrates failure rate of this particular structure, when applied to the full dataset, like this manner the portion of lower lip are detected in 28 images and this particular craniofacial structure failed detected in rest 2 images whether the upper lip is detected in 26 images and failed to detect on rest of 4 images, reported in the table below. In the case of detection of soft nose tissue profile the experimental result shows that this structure is detected absolutely in 22 images, but in rest of 8 cases the proposed methodology failed to detect the specific structure, table below illustrates the result. The nasion curve is detected in 23 cases, and in rest of the 7 cases the specific structure is filed to be detected. The desired soft basin curve and hard basin curve are detected in most of the cases; the success rate for detecting this structure is 97-98%, but we face some problem during the detection of pituitary fossa, the problem is that, it is tough to extract this particular curve, as one single parameter for the threshold of Canny's algorithm does not work for all the images as this is an ambiguous region, although we succeeded to locate the structure impeccably in 26 cases, but in rest of 4 cases it is failed to be detected. Table below illustrates failure rate of each desired curve when applied on test dataset.

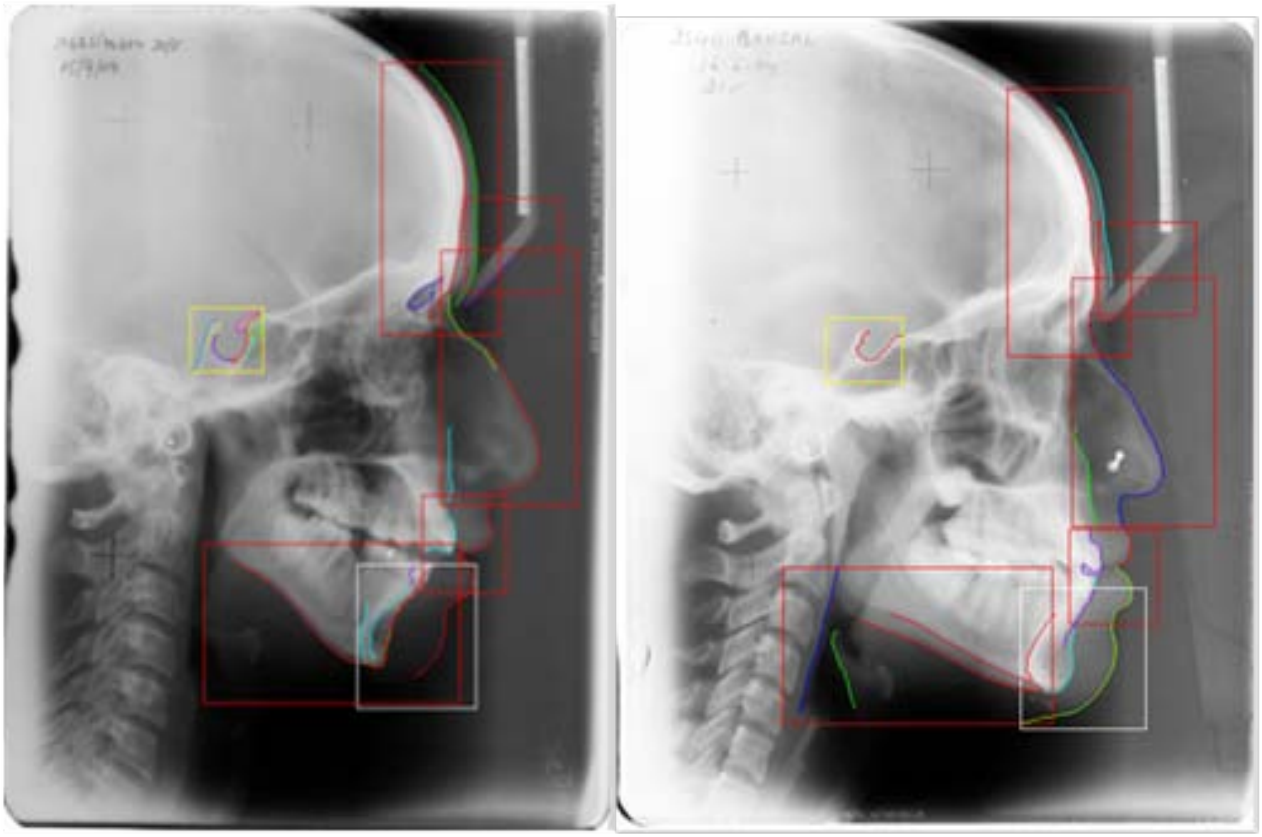
As the second testing phase, the algorithm tested on the dataset2, the analysis of the result shows that the detection of hard and soft tissue profile of chin succeeded in almost every image. The angle of mandible and its lower border are detected absolutely in 27 images but in rest of 3 cases it failed to detect the structure, table below depicts result. The portion of upper lip and lower lip are detected in 95-98% images. Soft nose tissue profile is detected absolutely in 27 images, but in rest of 3 cases it failed to detect the structure, table below illustrates the result. The nasion curve is detected in 95% of cases. The desired curve at the above regions of nasion i.e. soft basin curve and hard basin curve are detected in most of the cases; the success rate for detecting this structure is 98%, we succeeded to locate the pituitary fossa flawlessly in 21 cases, but in rest of 9 cases it is partially detected. Table below illustrates failure rate of each desired curve when

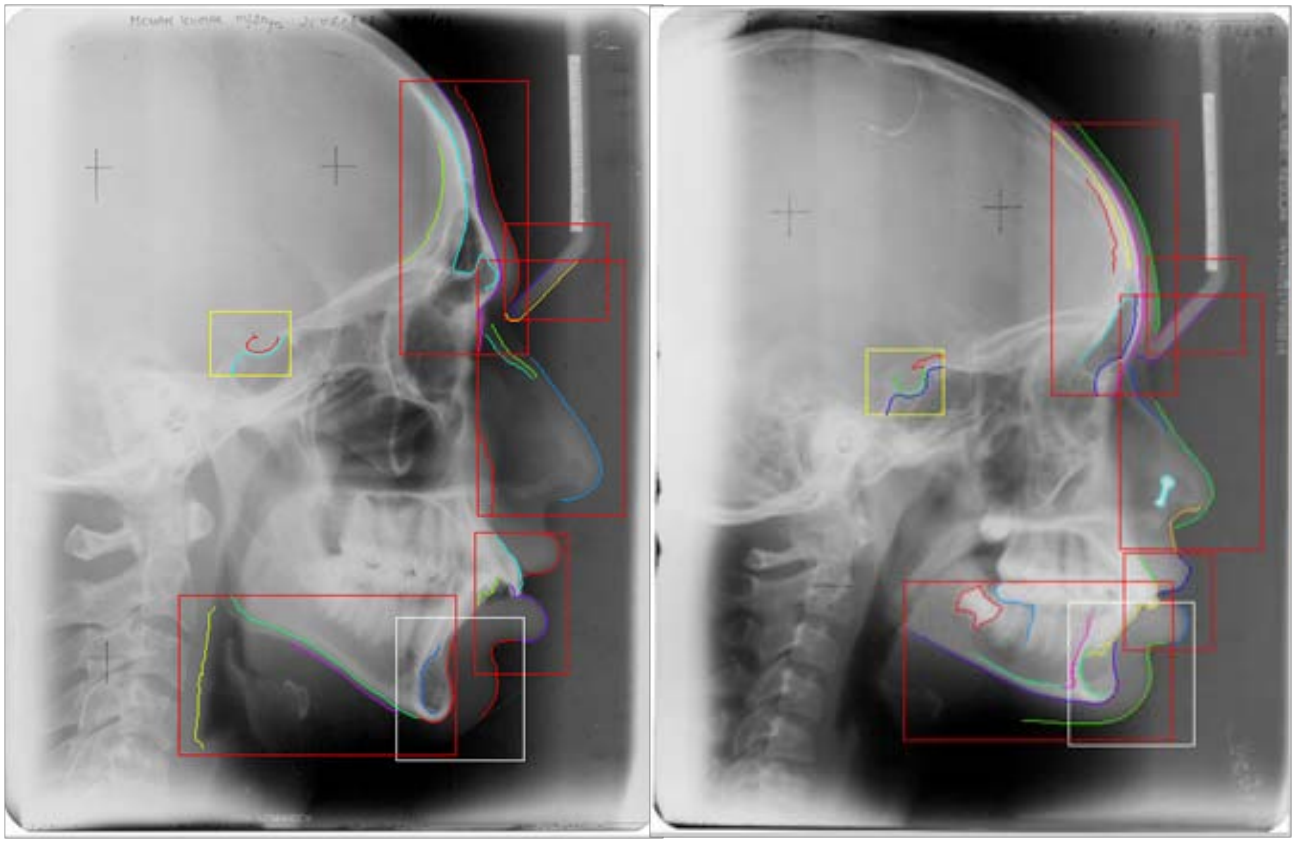
applied on test dataset. Some of the experiment results of the proposed algorithm are shown in the figure below.

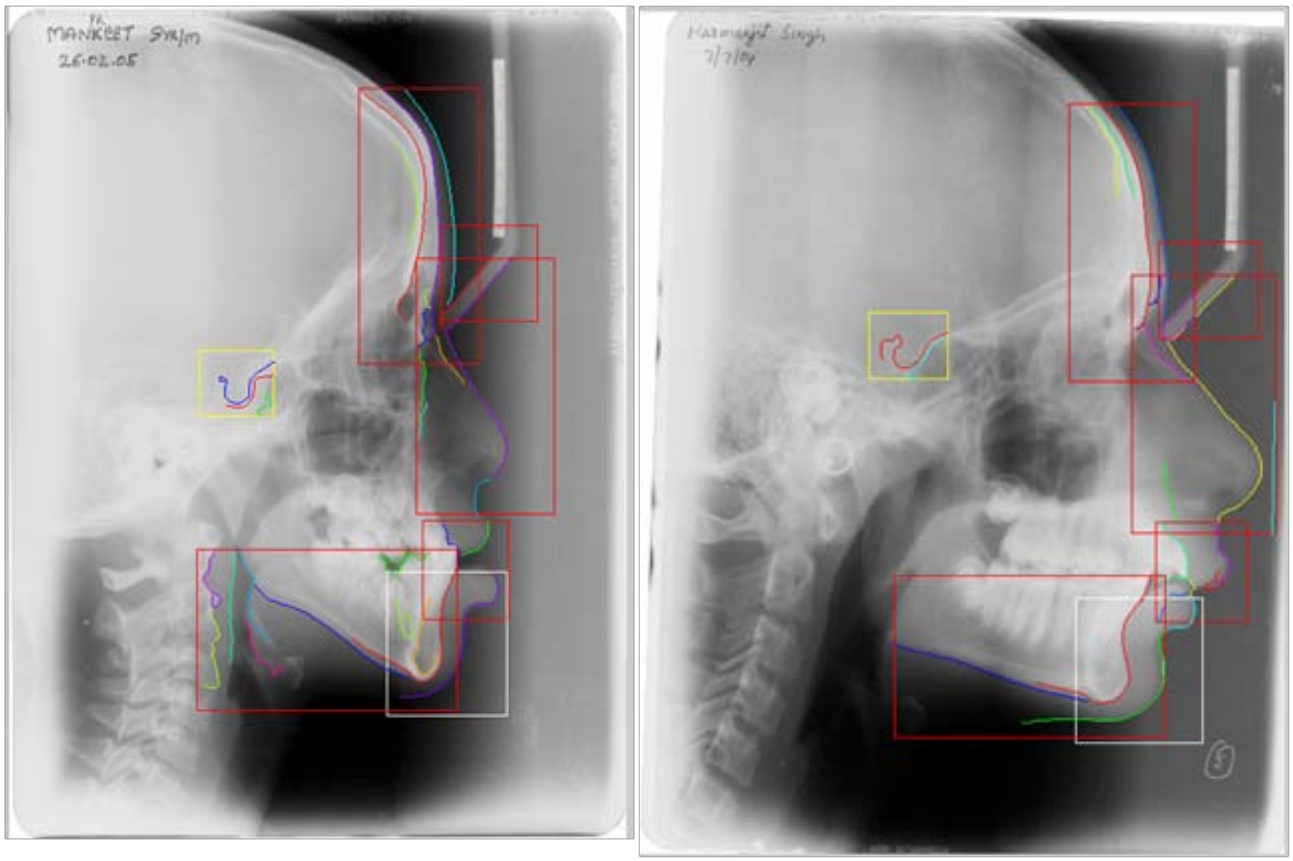
Name of the curve	**Failed (number of images)**	
	Dataset1	**Dataset2**
Soft tissue profile of chin	5	0
Hard tissue profile of chin	1	1
Angle of mandible and it's lower border	3	3
Lower lip	2	0
Upper Lip	4	0
Soft nose tissue profile	8	3
Nasion	7	0
Soft basin curve	2	0
Hard basin curve	1	0
Pituitary Fossa	2	4

Table 4.1: Illustrates failure rate of the algorithm when applied in dataset 1 and dataset 2








MANKEET SINGH
26.02.05

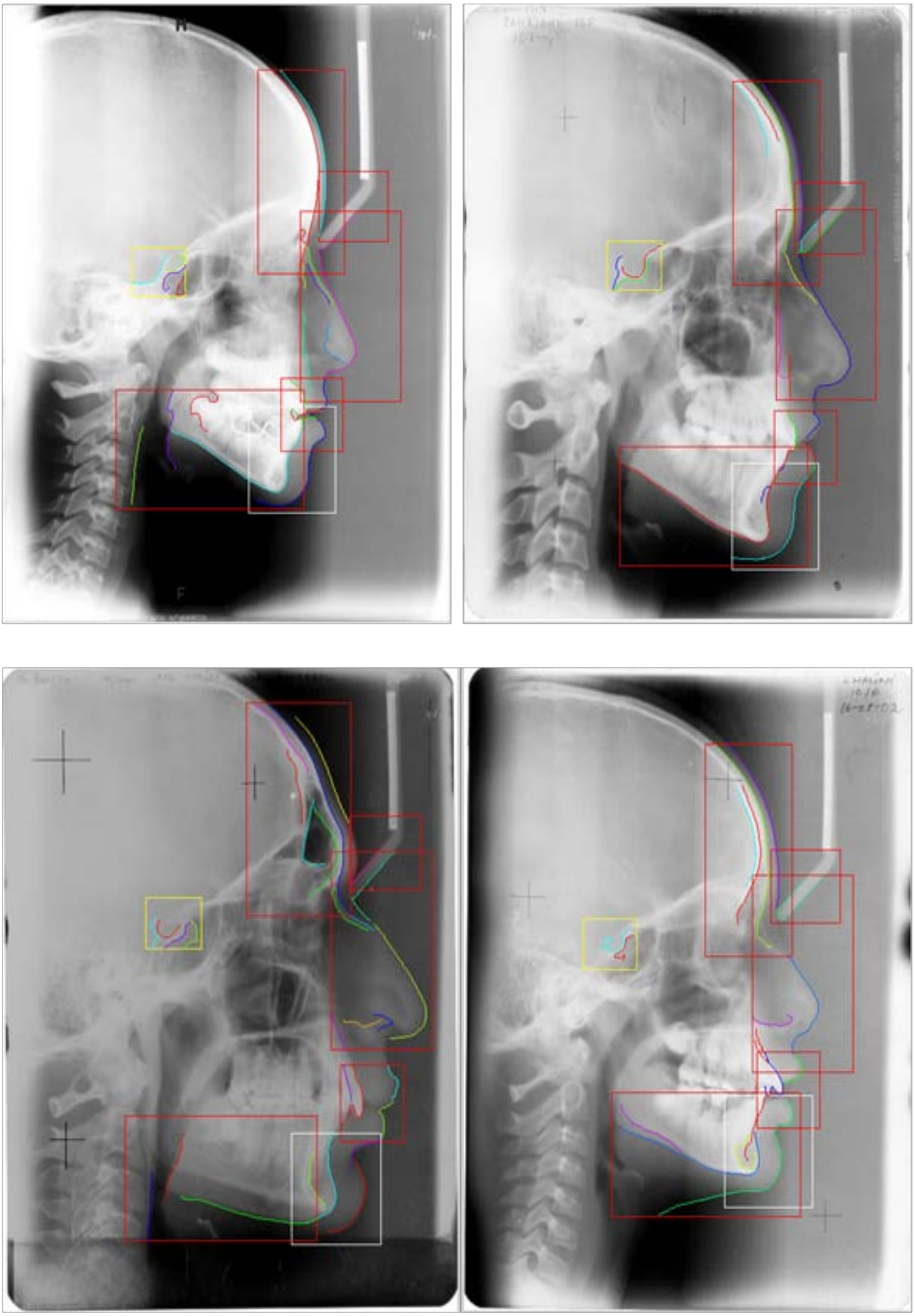
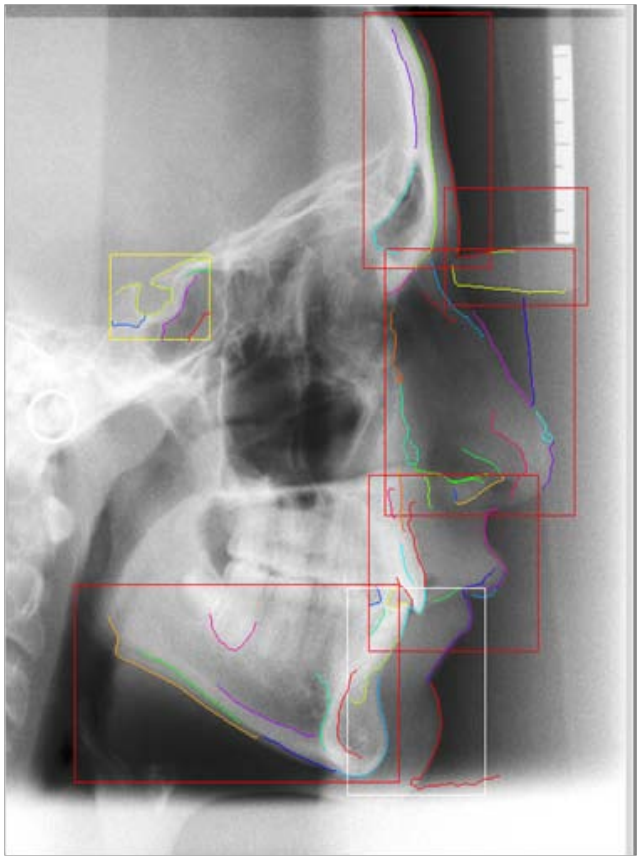
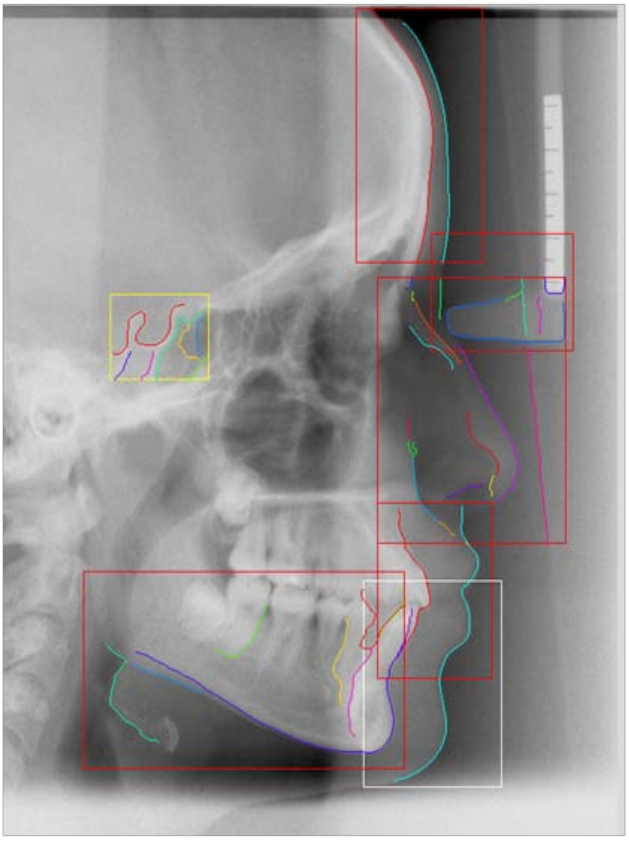
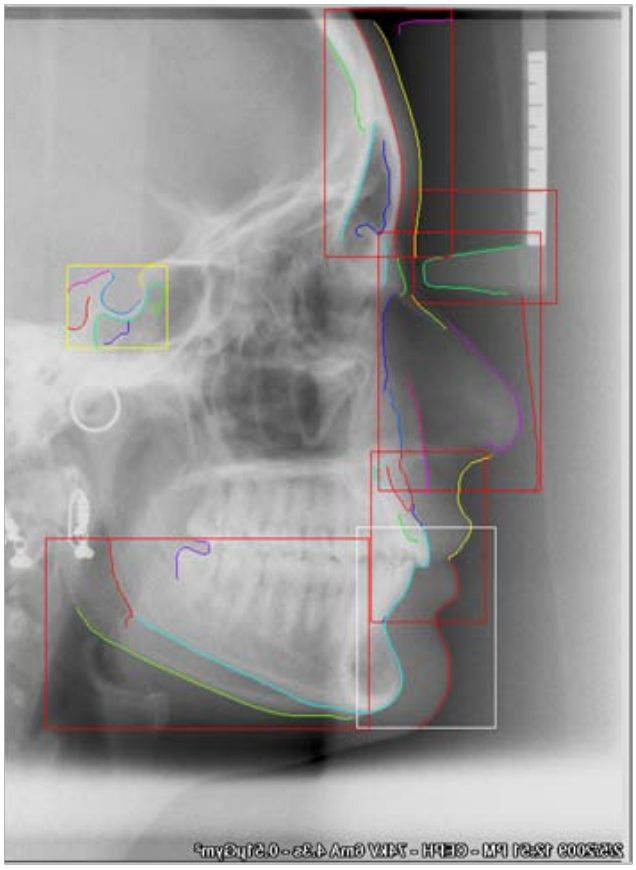
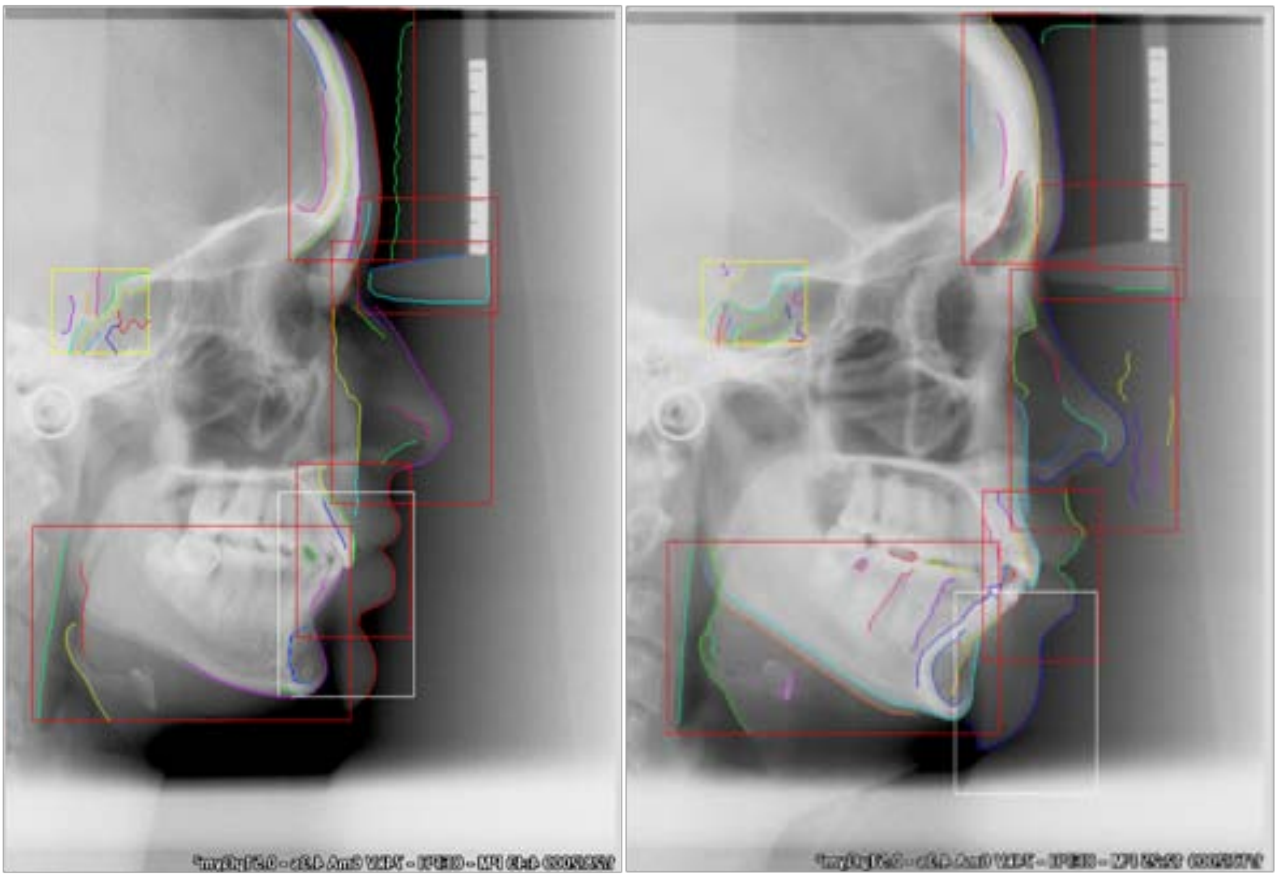


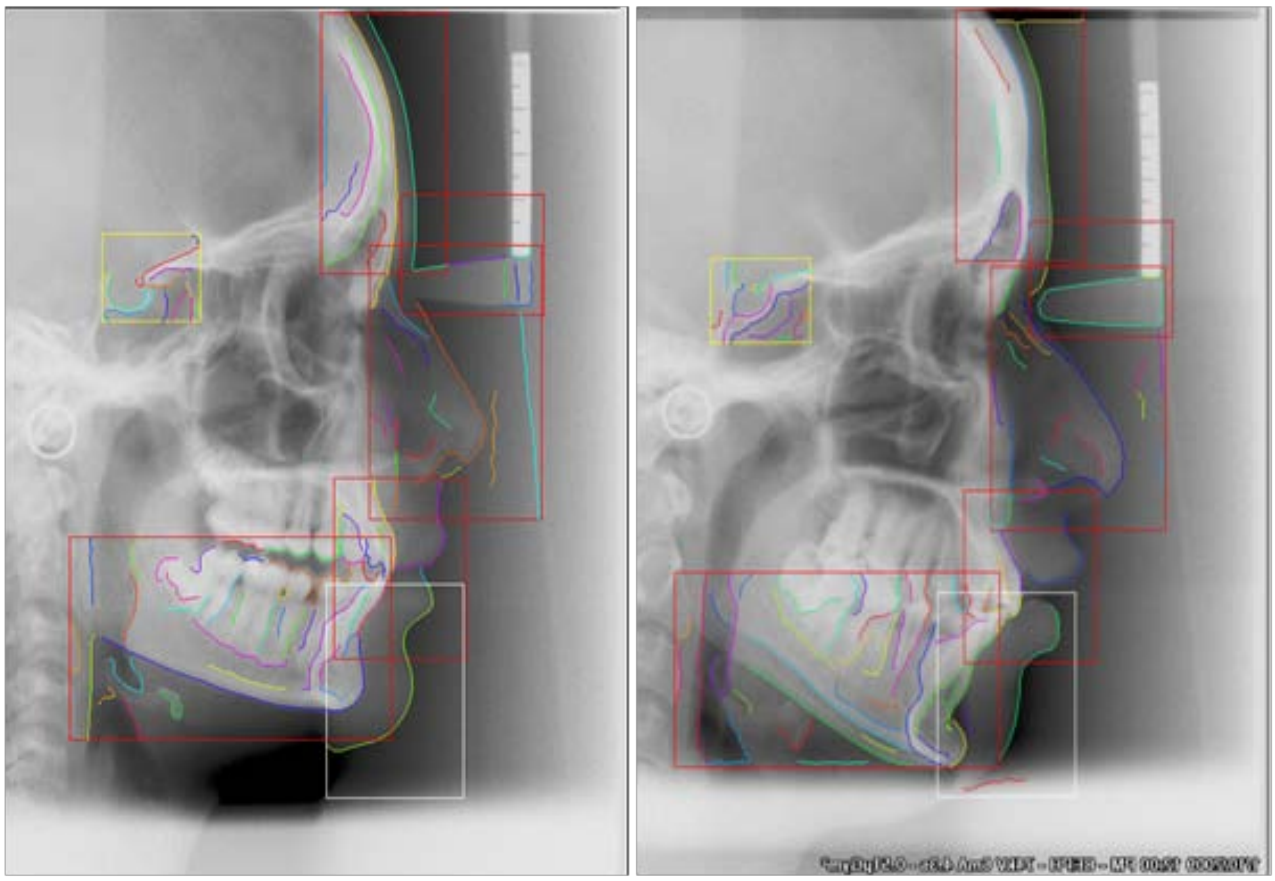
Fig 4.11 (a): Shows results of the proposed algorithm on the dataset 1











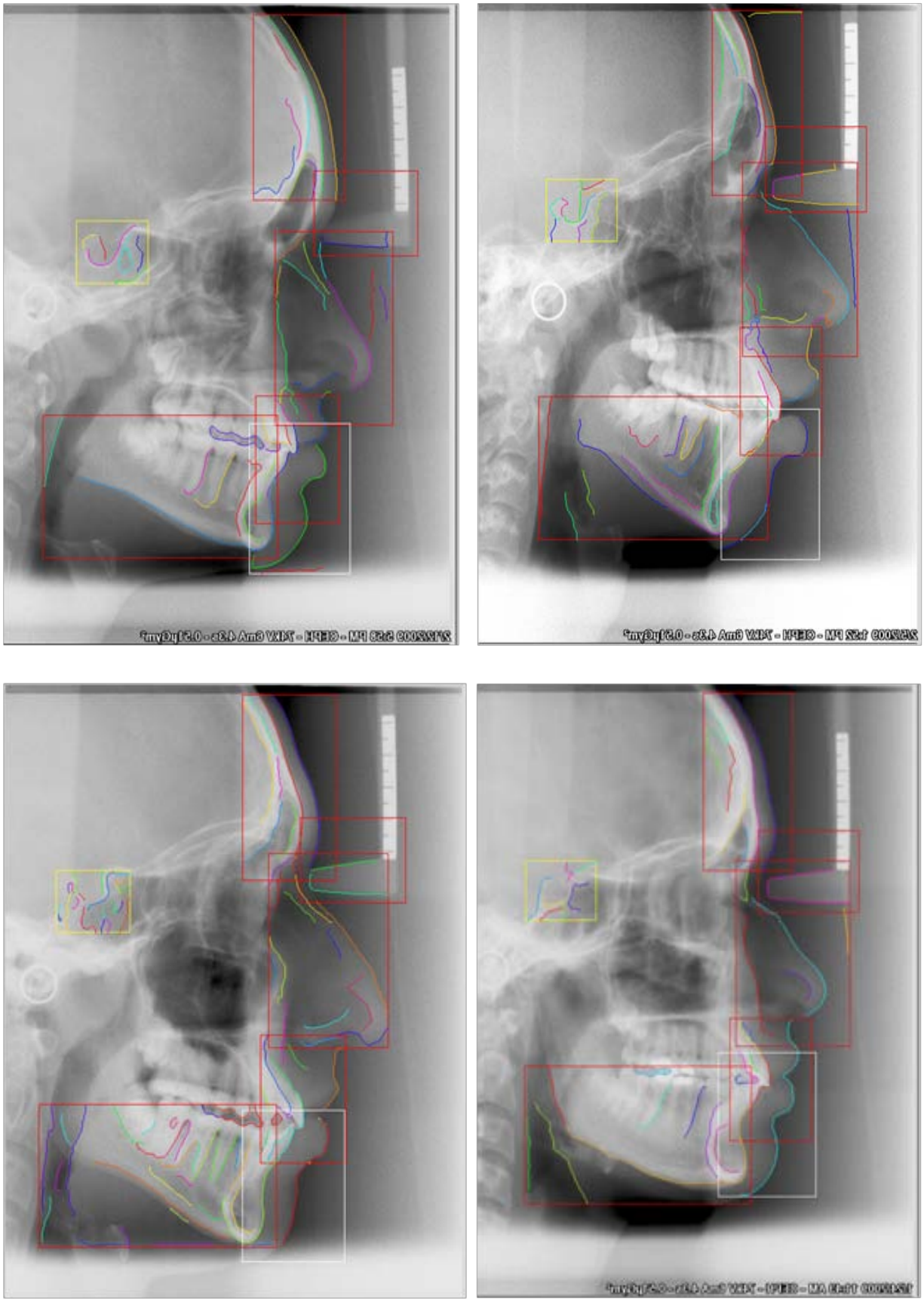


Fig 7: Shows results of the proposed algorithm on the dataset of 25 images.

CHAPTER: V

AUTOMATED CEPHALOMETRIC ANALYSIS BY MODEL BASED APPROACH

5.1 PROPOSED APPROACH

The combination of image processing technique and statistical model is proposed for feature point localisation. On cephalogram human experts locate landmarks by hand and the locations of their located co-ordinates are stored in a database. For each image, principal component analysis (PCA) is applied on the located landmarks for statistically characterize its shape variation and grey profile of each feature point. For an input sample, the reference landmarks are recognised by edge detection and template matching technique, and used to divide the input shape. Then the feature points of each input shape are located by a modified active shape model.

5.2 FEATURE EXTRACTION

For representing each image we extract some feature points from the image, this section presents the method for extracting some feature points from the image. Edge detection technique followed by identification of singular point on the edge, or pattern matching techniques which detect directly the structure where landmarks/feature points are located. Both approaches have disadvantages: Edge detection depends strongly on the image quality moreover, not all landmarks are located directly on the significant edge of the image, on the other hand pattern matching is a slow technique and has the possibility of giving false alarms, wrong detection of structures similar to landmarks, but located on distinct position. To overcome these disadvantages, we have devised a combination of the two approaches as the following modules that are applied in a manner that the algorithm first detects the region of interest for various landmarks followed by defining a small search area for each landmark, where edge detection is applied and then geometrical definition of the landmark is used to localize it. If the landmark is not on the edges then the geometric definitions of the landmarks are used to localize the landmarks. These steps are now explained in brief. As we are interested in the locations, it is desirable to convert the original image into binary image for the following reason. We can divide our algorithm of feature detection into some modules, described as follows:

5.2.1 REGION DETECTION MODULE

5.2.1.1 ENTRY POINT OR REFERENCE POINT DETECTION

The machine tripod rod is selected as reference region black square box (Fig. 5.1), as a very angular structure in the cephalogram images. This structure is present in a dark region and this region shows a high contrast hence results, in a good edge detection, which justifies the reason for choosing this as an entry region, and can be easily recognized by correlation based template matching technique. The search time is optimized by specifying the region (top right quadrant of the image). When the region of tripod rod is detected we enclose this location with a rectangle square box of same size as in template.

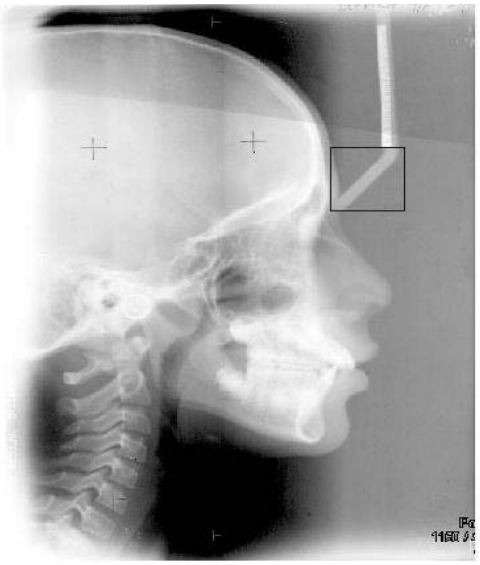


Fig :5.1 The entry region for an image

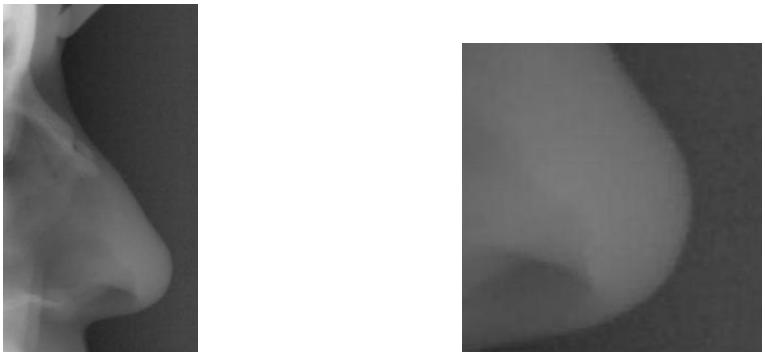
5.2.1.2 OTHER THREE REGION DETECTION MODULE

After detecting the initial point then with the reference of the co-ordinate of left lower corner of this box, we search near about regions with a template of nasion (fig:5.2(a)), we localised the region of nasion, after that we does a second level search with a more precise template in the region detected by the first level search (fig:5.2(b)), next we perform a third level search with another more precise template(fig:5.2(c)), after localization we perform the edge detection module described below, after that the nasion landmark is detected according to the geometrical definition of the of nasion, the nasion is defined by "the most anterior point of the nasofrontal in the median plane". For detecting the nasion ($\boldsymbol{p_1}$) the edge detection module followed by distance measure module (described below) are applied.

(a) (b) (c)

Fig 5.2: (a)shows the template of nasion for first level search,(b)template used for second level search (c) template used for third level search

By making the location of the first level search of nasion as the reference point for detection of the location of nose tip, the time complexity for searching the next template can be reduced and more précised, as the variation of the template in fig:4 are investigated for all the images and the range for searching for this convinced template can be fixed. As according to the geometrical definition of the human face the template of fig5.3(a) lies below the region of fig:5.2(a). The same approach with the nose template (fig:5.3(a)) for detecting the approximate region of nose. After detecting the approximate location, a second level search with a more precise template (fig:5.3(b)) in the region detected by first level search are applied. By application of the same edge detection approach described below and by applying the distance measure approach, the nose tip ($\boldsymbol{p_2}$) can be located, in that case we take minimum distance as the criteria for locating this landmark.



(a) (b)

Fig 5.3: (a)shows the template of nose for first level search,(b)template used for second level search.

Next based on the location of the first level search of nose region the region for location of chin is searched with template (fig:6) to find the approximate region of chin, next a second level search with a more precise template (fig:7) for detecting the précised location of the chin are applied. Then we apply the same approach of edge detection & distance measurement to pinpoint the

location of menton because according to the geometrical definition of Me ($\boldsymbol{p_3}$), the Me is defined by "the most caudal point in the outline of symphysis it is regarded as the lowest point of mandible and correspond to the anthropological gnathion " , so in that case the minimum distance are kept as a criterion for pointing out the location of the landmark.

(a) (b)

Fig 5.4: (a)shows the template of chin for first level search,(b) template used for second level search.

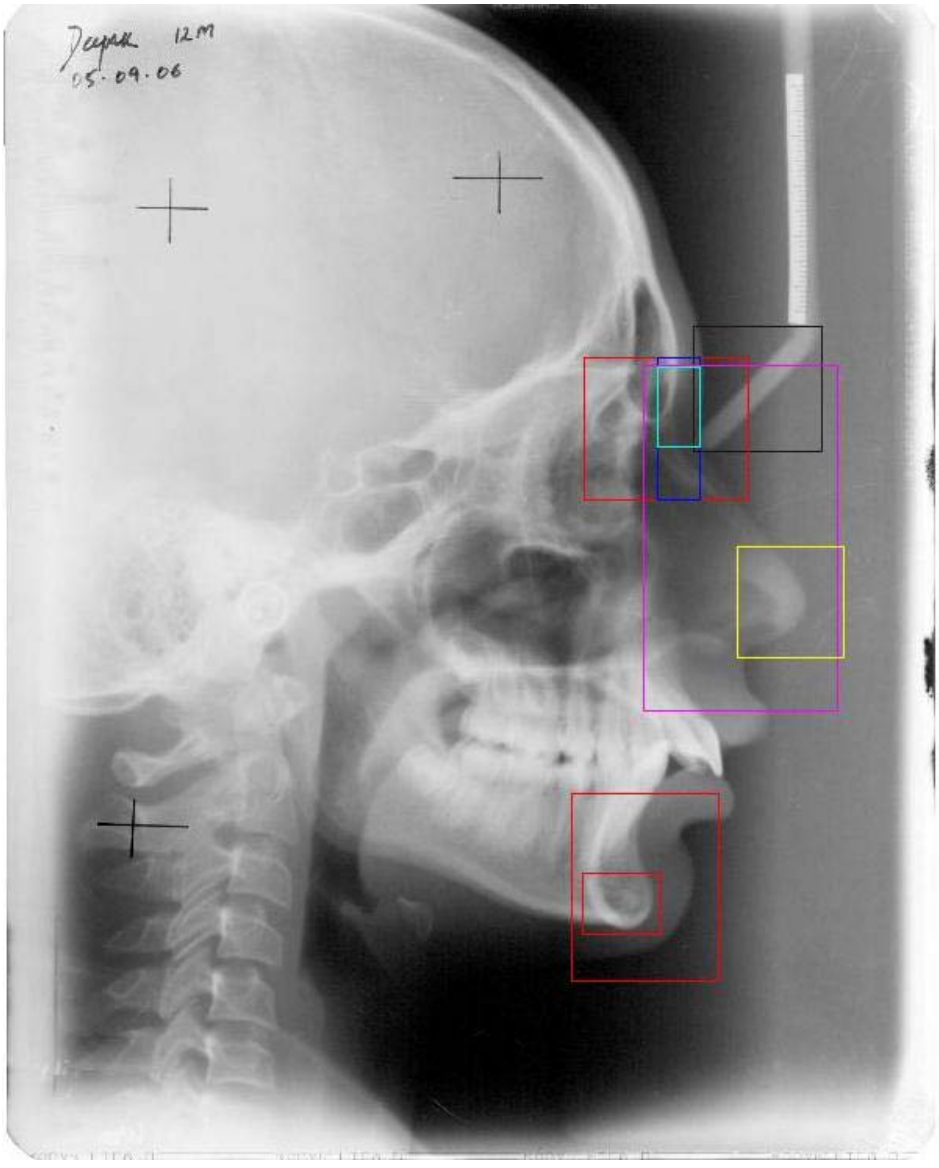


Fig 5.5: Results of template matching with the template described above

5.2.2 EDGE DETECTION MODULE : Typically X-ray images are low contrast images, so for enhancing the image quality Adaptive histogram -based equalisation can be applied to aid viewing the key features in the image. Figure (5.6(a)) shows before applying adaptive histogram -based equalisation and Figure (5.6(b)) shows after the operation on the portion of nasion. As local processing on image shows better result than the global processing, so along row wise the image is divided into five sub images. Then by applying a local threshold for each sub image the image are converted to binary. Threshold are fixed using "Otsu's" method, which chooses the threshold to minimize the intra class variance of the thresholded black and white pixel. Figure (5.6(c)) shows the result of this operation on nasion. Next by applying canny edge detection technique the edges are detected form the image. Figure (5.6(d)) shows the image of chin portion and Figure (5.6(e)) shows the result after edge detection. For linking the broken edges repeated morphological opening operation (Fig 5.7) are applied. Experiment shows that repetition of dilation and erosion operation with square structuring element (of width of 2 pixels) can improve result of edge map. For our case we repeat the morphological opening and closing operation for three times.

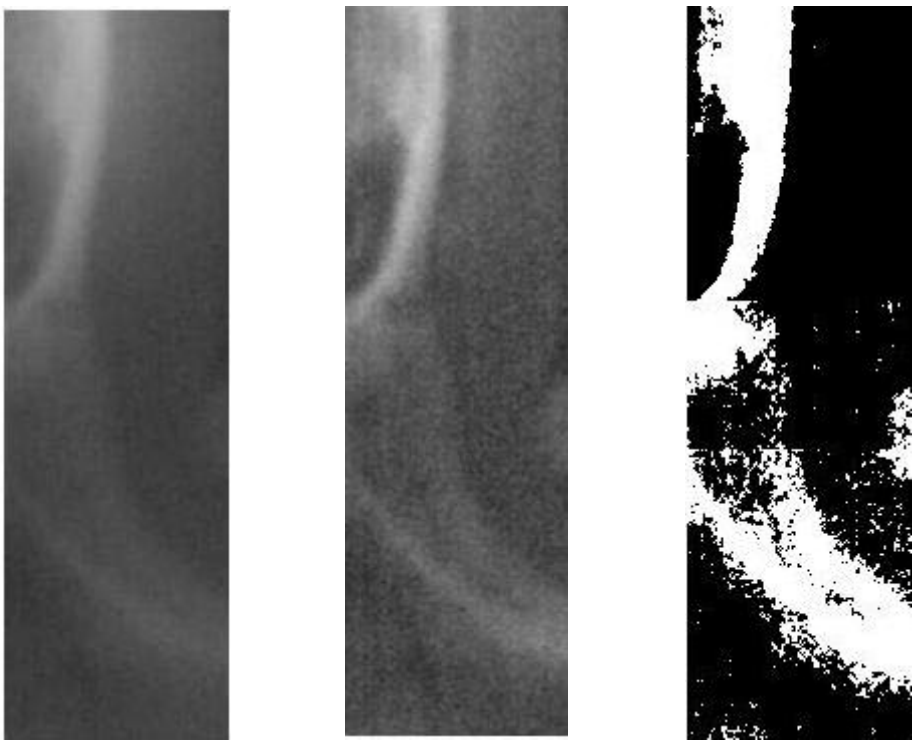


Fig 5.6: (a) Shows the original image (b) image after applying adaptive histogram equalization (AHE)(c)After applying thresholding operation (d)show the chin template (e)edge detected after AHE ,thresholding

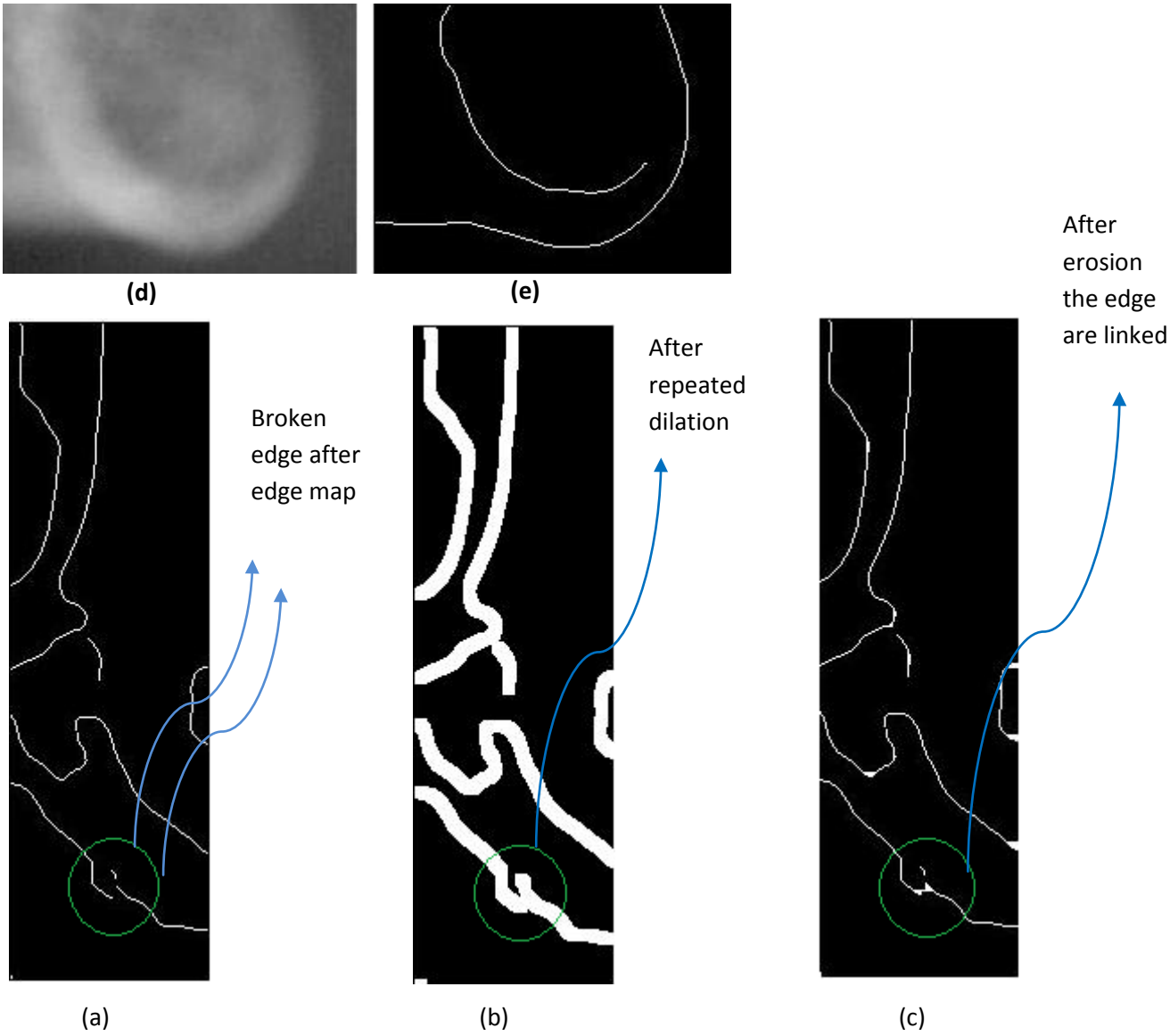


Fig 5.7:(a)Edge detection on the fig 5.6 (c). (b)See the broken edges are linked after repeated dilation (c) The broken edges are linked after repeated erosion , no. of repetition is equal to no. dilation

5.2.3 Distance measurement module : A raster scan is performed from the upper right side to the lower right side in the template until the first white pixel is found, and by counting the distance of the first white pixel for each case (figure:5.8(a),(b)) .

After detecting this three important point, some important features is calculated using their coordinates by making a shape (fig:5.9), so that more feature can be generated using the extracted three points. The created feature is lines measuring the distances between points and measuring the distances between those lines. The lines represent the scale and local shift of the skull relative to lower corner of the image. The lines are drawn to join the points p_1, p_2, p_3 and the centre (**c**) to obtain fig (5.9). Table 2 describes the features generated from the figure.

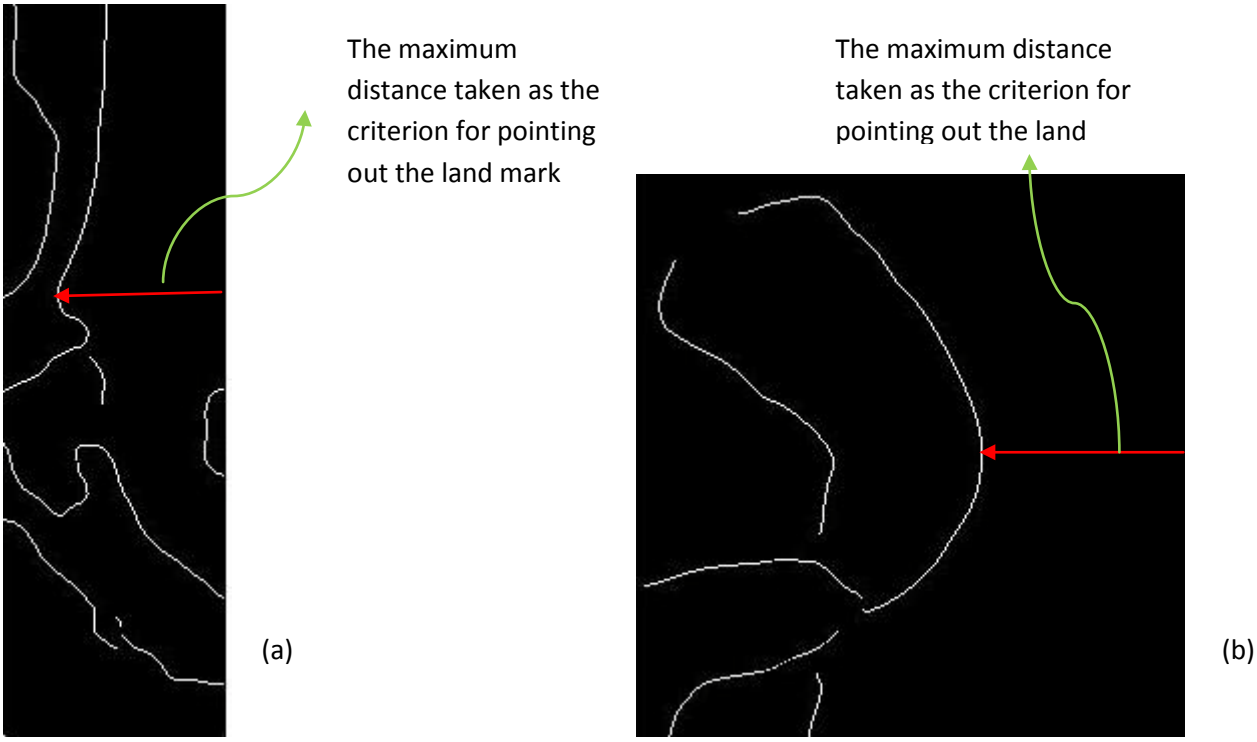


Fig 5.8 (a) The land mark detected by measuring the maximum distance (b)The landmark by measuring the minimum distance

As it is difficult to locate the other reference (expect some more which shows a bit error in localization) landmarks by template matching edge detection and distance measurement techniques, so pattern matching algorithm is proposed to locate them . As shown in Fig:5.9 the lines are drawn to connect the located landmarks and their centroid.

Name and description of the created features

No	Name	Description
1	Line1_ 2	Distance between P1 and P2
2	Line1_ 3	Distance between P1 and P3
3	Line1_ 4	Distance between P2 and P3
4	C_x	Horizontal coordinates of the centre of gravity
5	C_y	Vertical coordinates of the centre of gravity
9	c1	Distance between centre of gravity and P1

10	c2	Distance between centre of gravity and P2
11	c3	Distance between centre of gravity and P3
13	θ_1	Angle between Line1_2,c2
14	θ_2	Angle between Line1_2 and,c1
15	θ_3	Angle between c1&c2
16	θ_4	Angle between Line1_4 and c2
17	θ_5	Angle between Line1_4 and the c3
18	θ_6	Angle between c2 and the c3
19	θ_7	Angle between c3and the c1
20	θ_8	Angle between Line1_3 and the c2
21	θ_9	Angle between Line1_3 and the c3

Table 5.1 : Name and description of the created features

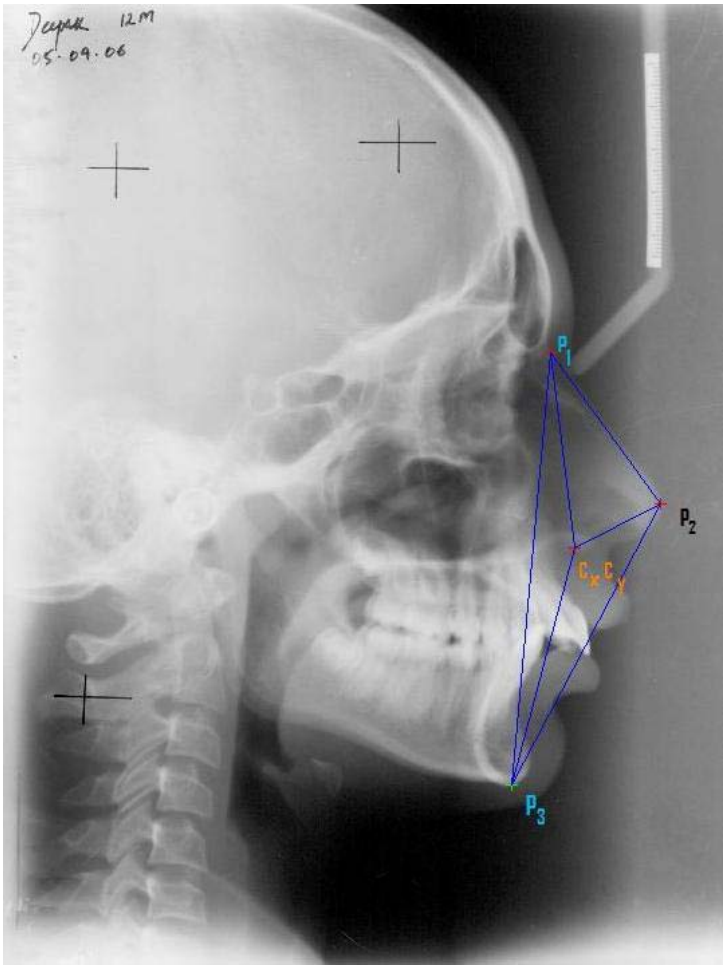


Fig 5.9: The landmarks are detected and features are measured between the landmarks

Centre of gravity is calculated by the following formula:

If the three points $P_1, (x_1, y_1) P_2(x_2, y_2), P_3(x_3, y_3)$. Then the centroid is represented by:

$$C_x = \frac{x_1 + x_2 + x_3}{3}$$

$$C_y = \frac{y_1 + y_2 + y_3}{3}$$

Then each x-ray mage can be represented by a 21 dimensional feature vector. This feature vectors are normalised by making Line 1_2 as the unit length and $\boldsymbol{\theta_1}$ as unit angle. thus the similarity between two images can be represented by the Euclidian distance between their feature vectors. For an input image, the top 5% most similar images in the training set are selected.

The next step of this algorithm is to align the selected training image to the input image by Procrustes analysis.

5.3 ALIGNING THE SET OF TRAINING SHAPES

The modelling method works by examining the statistics of the co-ordinates of the labelled points over the training set. In order to be able to compare equivalent points from different shapes, they must be aligned in the same way with respect to a set of axes. The required alignment by scaling, rotating and translating the training shapes so that they correspond as closely as possible. An iterative method which can minimises a weighted sum of squares of distances between equivalent points on different shapes.

5.6 ALIGNING SHAPE REGION

In order to study the variation of each feature point in the craniofacial structure throughout the training images the structure in each images are aligned to each other. i.e. suppose we have **k** feature points which represent the whole structure in an image .we can represent the landmark or feature point location in the i^{th} training shape by its corresponding feature vector:

$$\boldsymbol{X}_i = [x_i^0, y_i^0, x_i^1, y_i^1 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots x_i^{k-1} y_i^{k-1}]$$

Alignment of these shape is done by procrustes analysis, which is frequently used in medical image analysis [34].

Aligning a shape A to another shape B means:

- Scaling A so that the distance between two certain points will be same in case of A & B
- Alignment of their centroid .
- Rotating each shape so that the line joining two certain points is directed in a certain direction.

There exists some analytical solution [35] for aligning a set of shapes, the following approach suggest in [36] is sufficient.

1. The first step is chosen as the estimate of the mean shape.
2. Align all the remaining shape to the mean shape.
3. Again calculate the estimate of the mean shape $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$.
4. If the mean estimate has changed return to step 2.

This iteration will be recurring until the mean shape does not change significantly within iteration. Bookstein suggests that two such iteration is sufficient in most cases. In practice we iterate four

times and the mean shape does not change any more. In our method the original ASM is modified. As for each region, we estimate its mean shape as initial model $\mathbf{X}$ and map it from the model space to the image space by affine transform $\boldsymbol{X = T(x)}$**.** (Described below) Since the position of the pointed landmarks are correlated to the current region are known both in the model space and the image space

5.7 CAPTURING THE STATISTICS OF A SET OF ALIGNED SHAPES

Once the N aligned shapes is available the mean shape and variability can be found as follows:
Let x be a vector describing the n points of the i^{th} shape. The mean shape $\bar{\mathbf{x}}$ is calculated using:

$$\bar{\mathbf{x}} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{x_i} \qquad (1)$$

The way in which the points of the shape move together, it is known as modes of variation can be obtained by applying principal component analysis to the result of deviation from the mean as follows [11]. For each shape in the training shape it's deviation from the mean dx_i is calculated as follows.

$$dx_i = x_i - \bar{x} \qquad (2)$$

From this we can calculate the 2n×2n covariance matrix, S using

$$S = \frac{1}{N} \sum_{i=1}^{N} dx_i dx_i^T$$

p_k defines the modes of variation of the points of the shapes (k=1,2,.....2n) , which is the unit Eigen-vector of S such that.

$$Sp_k = \lambda_k p_k$$

(Where λ_k is the k' th Eigen value of s, $\lambda_k \geq \lambda_{k+1}$)

$$p_k^T p_k = 1$$

It can be shown that the Eigen vector corresponding to the largest Eigen values describe the most significant modes of variation in the variables used to derive the covariance matrix, and this will be the proportion of the total variance explained by the each eigenvectors is equal to the corresponding Eigen value [37]. Generally most of the variation can usually be describe by the small number of modes, t (<2n). The t can be calculated by choosing the smallest number of modes such that the sum of their variances explain a large proportion of λ_T , where λ_T is the total variance of all the variables,

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k$$

The k^{th}eigenvector affects point l in the model by moving it along a vector parallel to (dx_{kl}, dy_{kl}), which is obtained from the l'th pair of elements in p_k,

$$(dx_{k0}, dy_{k0}, \ldots\ldots, dx_{kl}, dy_{kl}, \ldots\ldots dx_{kn-1}, dy_{kn-1}) \quad (7)$$

Combinations of vectors, one for each mode, move the modeled landmark points around in the regions of the 'clouds' of scattered points from the aligned training set. Any shape in the training set can be approximated using the mean shape and a weighted sum of these deviation obtained from the first **t** modes.

$$x = \bar{x} + pb \quad (8)$$

Where $p = (p_1, p_{2,\ldots\ldots\ldots\ldots} p_t)$ is the matrix of first t Eigen vectors,

$b = (b_1, b_{2,\ldots\ldots\ldots\ldots} b_t)^T$ is a vector of weights, one for each Eigen vectors,

The Eigen vector are orthogonal, $p_k{}^T p_k = I$ so

$$b = P^T(x - \bar{x})$$

The above equations allow us to generate new examples of the shapes by varying the parameters **b** within suitable limits, so the new shapes will be similar to those in the training set. The parameters are linearly independent, though there may be non-linear dependencies still present. The limits for each element of b, b_k are derived by examining the distributions of the parameter values required to generate the training set. If Gaussian distributions are assumed one can choose sets of parameters $\{b_1, \ldots\ldots\ldots\ldots\ldots\ldots b_t\}$ such that the mahalanobis distance (D_m) from the mean is less than a suitable value, D_{max}

$$D_m^2 = \sum_{k=1}^{t} \frac{b_k^2}{\lambda_k} \leq D_{max}^2 \quad (10)$$

5.8 MODELING GREY LEVEL APPEARANCE

We can improve our initial estimation of landmarks locations by modeling the appearance of an object by checking the grey levels in the regions around each of the level model points in the training images. Considering a model point specifying a part of an object in a image, the grey level pattern about the corresponding point in images of different example will often be similar. We wish

to use our models for locating examples of objects in new images. The work of Bailes and Taylor [38] suggests that the location of model points in images can be improved by incorporating each point's grey-level environment into the model.

Although in general we could consider a region of any shape around each point, we will concentrate on one-dimensional profiles normal to curves passing through the points. This requires that we define the connectivity of the model points; in many cases this is straightforward, particularly when the points lie around a boundary. During the image search we wish to locate the best position of each model point so we find the area near to the current position of the model point. In order to achieve this we have to orient each modeled point of our shape model , in order to align it correctly. So it can be achieved by performing an orientation with respect to the nearby model point.

In some cases it is efficient to assume that the points lie on strong edges and to search for such in an image. However this is not always satisfactory and it is necessary to have a more general model of the grey-level appearance. For every model point **I** in each image, **j**, we can extract a length of the pixels profile, centered at the point. Following Bailes and Taylor [38], we choose to sample the derivative of the grey levels along the profile in the image and normalize. This gives invariance to uniform scaling of the grey levels and the addition of a constant, for each model point in the training image we can extract a grey level profile, g_j, of length n_p pixels, centered at that point. Where p_{start} the start point is p_{end} is the end point of the profile, where the profile is of n_p pixels, the k^{th} element of the profile is

$$g_{jk} = I_j(y_k)$$

Where y_k is the k^{th} point along the profile.

$$y_k = p_{start} + \frac{k-1}{n_p - 1}(p_{end} - p_{start})$$

Here $I_j(y_k)$ is the grey level in image j at that point.

We can calculate a mean profile $\bar{g}$ for each model point and an $n_p \times n_p$ covariance matrix, S_g is the statistical description of the expected profiles at the point. A compact model of the allowable variation in the grey level profiles can be produced by applying the principal component analysis as it was for flexible shape models.

P_g is the variation about mean. The Eigen vector of S_g corresponding to the largest Eigen values $t_g (\leq n_p)$. By analogy with (8) we can write

$$g_{new} = \bar{g} + P_g b_{gnew}$$

Where b_{gnew} is the set of t_g parameter describing the profile model.

Suppose we want to measure how well a profile, g extracted from the image, fits a model for a point, represented by $\bar{g}, P_g$ and the t_g largest Eigen values λ_j , j =1.....t_j The parameter required to best fit the model is given by

$$b_g = P_g^T (g - \bar{g})$$

The best fit to the model g is then given by (11)

$$g_{best\_ft} = \bar{g} + P_g b_g \tag{11}$$

If all the eigenvector were used in the model ($t_g = n_p$) then this would be exactly equal to g. However typically this expansion is truncated ($t_g < n_p$) there will be a difference between the two. The sum of squares of differences, R^2, is given by;

$$R^2 = (g - g_{best\_ft})^T (g - g_{best\_ft})$$

It can be shown that

$$R^2 = (g - \bar{g})^T (g - \bar{g}) - b_g^T b_g \tag{12}$$

The measurement of the quality that how well the model fits the profile can be given by Mahalanobis distance.

$$M = \sum_{j=1}^{n_p} \frac{b_{gj}^2}{\lambda_j} \tag{13}$$

Where λ_j is the Eigen value to the corresponding j^{th} Eigen vector, where $\lambda_j \geq \lambda_{j+1}$

If $t_j < n_p$ we cannot calculate b_{gj} for $j > t_g$, we have $\lambda_j \leq \lambda_{t_g}$ for $j > t_g$, if we make the approximation that $\lambda_j = 0.5\lambda_{t_g}$ for $j > t_g$ then

$$M \approx \sum_{j=1}^{t_g} \frac{b_{gj}^2}{\lambda_j} + \frac{2}{\lambda_{t_g}} \sum_{j=t_g+1}^{n_p} b_{gj}^2$$

It can shown using (12) that

$$\sum_{j=t_g+1}^{n_p} b_{gj}^2 = R^2$$

Thus the measure that how well the model fits the profile is

$$F = \sum_{j=1}^{t_g} \frac{b_{gj}^2}{\lambda_j} + \frac{2R^2}{\lambda_{t_g}}$$

Where F approaches zero as the quality of the fit is improves.

If the distribution of the original profile is to be normal, the probability that g comes from the same population is approximately

$$p(g|model) \propto e^{-p}$$

The method above can be applied either to the raw grey levels or the derivative of the grey levels along the profile in the image. Also we can normalize the profiles prior to training and evaluation using

$$g'_{ij} = \frac{g_{ij}}{\sum_{k=1}^{n_p} |g_{ijk}'|} \qquad (14)$$

The constraint of $|b_g| \leq 3\sqrt{\lambda_g}$ sis applied to b_g , since it is assumed that the variations of training shape are distributed as a multivariate Gaussian on p_g. With the newly obtained shape parameter , $\boldsymbol{x}$ is updated according to (13) and $\boldsymbol{X}$ is updated according to (14) in the image space. The grey matching and the parameter update are iterated until $\boldsymbol{X}$ is converged.

5.9 EXTENSION TO MULTI-RESOLUTION IMAGES

The above said approach works every well for many different images, but the main problem occurs during the choosing of the length of the profile, because the length should such that for which the model landmark will be best fitted to the new location. If the profile becomes too short then the model landmark point will be too close to their targets in the image before they can latch on or pull the shape model into the place. If the search become too long then it will be computational expensive and the grey level models are more likely to latch on to distracting structures in the image away from the target object that prevents ASM from converging to the correct place.

Ideally the search will first look far from its current location and make a large jumps. Then it will come back to its home location, it will restrict its examination of the image to the immediate locality when it comes close to its desired target location. This suggests a multi resolution approach, with the models applied first to a coarse, low resolution version of an image, then refine on a higher resolution version. We can generate such images from an original image using

Gaussian smoothing and sub sampling to produce a multi resolution pyramid [41]. The original image is at the pyramid's level 0 location. Level 1 in an image is the half the number of pixel of the image along each axis.

In this implementation each pixel in the given level is generated by smoothing the image at the level below by 5×5 Gaussian filter, which linearly decomposed into two 1-5-8-5-1 convolutions and then the sub sampling of every other pixel.

In order to apply an ASM at a particular level in the pyramid grey level models trained on the data at that level are required because Gaussian smoothing and sub sampling significantly modify the image structure. Instead of a single grey level model for each landmark we generate a set of models one for each level of pyramid we wish to use. we wish that our model will use the same number of pixels at each level, so those at level l+1 cover twice as much the image area as those at level l (fig:5.10). Thus each grey level model at the coarser level may cover large part of structure of interest, while at fine level each model is much more localized. On a given layer , we need only search along a short profile sampled from the image. For the coarse resolution this will allow large movements for fine resolutions only small movement is allowed, if this procedure are applied then the grey level models are less likely to be distracted by image features away from the current image structure.

In order to perform local search ,we start from the highest level of pyramid and run a number of iterations of the ASM using the models trained at the level. During each iteration we look in the region around each point for a better location. We want to examine the image data at a small number of nearby locations, for instance at the current point and n_l pixel either side (figure 5.10). Having run a number of iterations at one resolution, we move to the next level down in the pyramid, and search at a finer resolution. Again we only need look n_l pixels either of side of each current point. This process is repeated until a suitable number of iterations are run on the original image at level 0 in the pyramid.

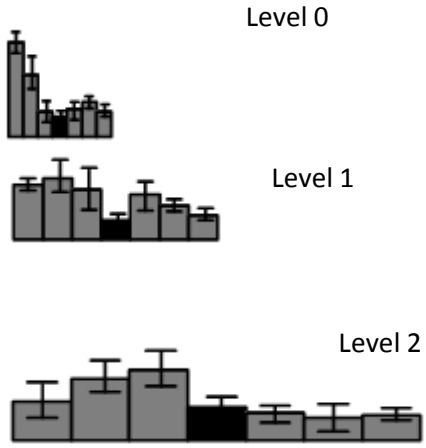


Fig : 5.10(a): Grey-levels models for a landmark point at different levels of the Gaussian pyramid

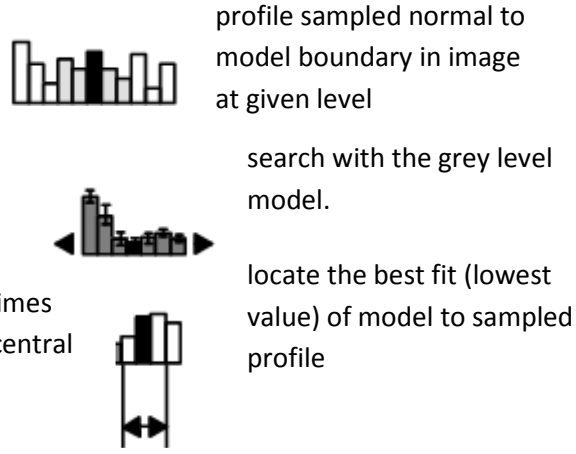


Fig:5.10(b) : At each level and each landmark the grey-level model fit at a small number of position about the current point is calculated, and number of best fit's lying in a central regions in recorded

5.10 CHOICE OF WEIGHTS FOR POSE AND SHAPE PARAMETER ADJUSTMENT CALCULATION

In the calculation of the adjustments to both the pose and shape parameters we can give a weight to each point to indicate the confidence we have in the suggested new position for the point. If all weights are set to unity simplifications can be made which reduce the complexity of the calculations required (Equation 21).

We have found that decreasing the weight for points which are found to be further away from the current model points than average, and may be outliers, works well:

$$w_i = \frac{1}{2 + |dX_i|^2} \qquad (17)$$

5.14 UPDATING THE POSE AND SHAPE PARAMETERS

By applying the equations above changes to the pose variables can be calculated. i.e.$dx_c, dy_c, d\theta, ds$,and the adjustments to the shape parameters db required to improve the match between an object model and image evidence. When these changes are applied we can ensure that the model only deforms into shapes consistent with the training set by placing limits on the values of b_k.As mentioned above, a shape can be considered acceptable if the Mahalanobis distance D_m is less than a suitable constant D_{max} , for instance 3.0 (See Eq. 9). That is to say, the vector b should lie within a hyper-ellipsoid about the origin. If updating b leads to an implausible shape, i.e. $D_m \geq D_{max}$ and the point lies outside the ellipsoid, b can be re-scaled to lie on the closest point of the

allowed volume using $b_k \rightarrow b_k \frac{D_{max}}{D_m}$ (k=1.....t)

Note that we have already applied implicit limits of zero to the weights of the eigenvectors truncated from our representation (i.e. $b_i = 0 \; \forall \; i > t$). Once the parameters have been updated, and limits applied where necessary, a new example can be calculated, and new suggested movements derived for each point. The procedure is repeated until no significant change results.

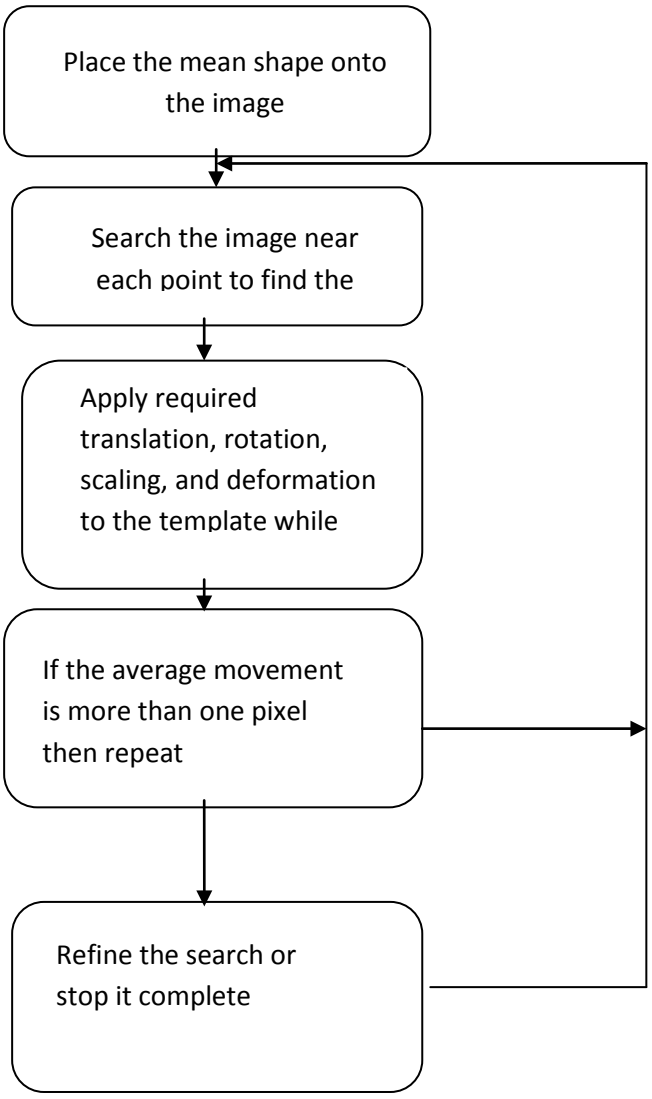


Figure 5.13: Flow chart for active shape model

5.15 RESULTS:

Figure (5.14) shows the landmarks are detected on the tested image after all the training examples have been aligned. It should be noted that this corresponds with the informal notion of the average person. Although this average is for small amount of examples in the training set but the method could be used to find out the means over larger populations. For testing the accuracy of the algorithm a drop one out method was used in the training set (Roudolph et. al.,1998), i.e. each image was removed from the total set and the model recomputed before testing, so the model was based on 3 (5% of 63-1) images and this model are used for training. This method of training is statistically acceptable, makes maximum use of training set and provides maximum number of tested images. Result obtain by the algorithm are compared with those obtained by the human experts. Before the experiment human experts located the landmarked on every testing images According to the clinical requirement, if the difference between the location of a feature point obtained by the algorithm and the criterion is less than 2 mm, the result is considered to be correct; if less than 4 mm, it is acceptable. If the rate of unacceptable feature points on a image is more than 20%, the failure of the algorithm on this image is declared.

The methodology failed to correctly locate the feature points in some cases after the adjustment of the model points by the grey profile matching, the reason is the misplacement in the grey profile matching. Because in our study there are a group of feature points that have similar grey profiles because of the dense distribution of the feature points and the nature of cephalogram, the grey matching therefore, is not convergent or is error prone. The misplacement will increasingly occur when the shape similarity between the shape model and the input shape is low.

In the next mode, by permitting interactive modification of the by users, the located feature points are placed in a better location, after that grey profile matching approach are applied, the result demonstrates that the algorithm failed in 8 cases, and the recognition rate improves correspondingly, the iterative grey matching make the feature points tend to be more closer to the right location. Table below demonstrates obtained result, when the algorithm was applied on the testing dataset.

LAND MARKS	AVARAGE(MM)	MAXIMUM(MM)	MINIMAM(MM)
Porion	12.70884	26.63163	0.692641
Sella	13.60579	47.65316	2.034117
Nasion	17.44279	37.85247	1.433779
Orbital	15.15666	31.11618	3.319293
UIR	16.69103	39.10698	3.4147
UIT	18.78269	40.90109	3.202276
Gonion	17.31599	82.96665	1.867303
menton	23.78635	63.52269	0.870675
Gnathion	23.01441	45.36045	1.313613
Pogonion	21.36162	43.59449	2.84369
Bpoint	20.59041	48.40539	4.539251
PNS	15.48302	61.53459	0.671665
ANS	16.52902	35.86843	3.339616
Apoint	16.81453	36.82068	4.093768
LIR	18.5004	43.61493	1.089741
LIT	17.93305	40.97997	1.924173

Table 5.2: Landmark errors across the tested images obtained before the application of grey profile matching approach for more précised localization, without permitting user to do the interactive modification of the detected position of reference landmarks

LAND MARKS	AVARAGE(MM)	MAXIMUM(MM)	MINIMAM(MM)
Porion	20.19322	37.00311	4.268338
Sella	16.88488	44.43863	2.470794
Nasion	16.09668	61.54834	1.393699
Orbital	20.45672	61.11185	1.422505
UIR	17.03298	51.23775	1.224866
UIT	20.7467	55.74099	2.144102
Gonion	23.1268	113.0914	1.882474
Menton	22.11421	66.36141	0.364526

Gnathion	21.42264	49.60545	0.630677
Pogonion	22.42999	47.09542	0.722413
Bpoint	21.14293	57.84479	0.403676
PNS	14.70113	33.07204	3.176026
ANS	20.33663	56.89856	2.210291
Apoint	19.25146	57.8741	2.162468
LIR	23.30628	60.46915	1.076496
LIT	19.94633	50.55134	3.370275

Table 5.3: Landmark errors across the tested images obtained after the application of grey profile matching approach for more précised localization, without permitting user to do the interactive modification of the detected position of reference landmarks

LAND MARKS	AVARAGE(MM)	MAXIMUM(MM)	MINIMAM(MM)
Porion	9.825366	18.78022357	5.555524
Sella	8.558773	14.25590312	2.65501
Nasion	12.96045	14.97074609	9.333
Orbital	9.146546	11.2323608	5.664141
UIR	6.574587	10.76436008	3.635388
UIT	7.382213	18.53409769	2.8261
Gonion	8.379567	11.4713523	5.003325
Menton	8.258837	19.54001094	2.81773
Gnathion	8.378366	19.64021741	3.116506
Pogonion	8.991398	19.75741907	3.483462
Bpoint	7.884867	15.44939389	4.354599
PNS	6.834037	9.647722453	4.033197
ANS	5.972022	11.338246	1.175644
Apoint	5.563921	11.01833316	1.981828
LIR	7.070512	12.60443532	2.380759
LIT	6.87516	14.94018226	3.928945

Table 5.4: Landmark errors across the tested images obtained before the application of grey profile matching approach for more précised localization, through permitting the interactive modification of the detected position of reference landmarks by user

LAND MARKS	AVARAGE(MM)	MAXIMUM(MM)	MINIMAM(MM)
Porion	17.77864326	31.73584	3.47563
Sella	16.45034223	27.60553	8.834694
Nasion	17.97763682	31.89293	4.840236
Orbital	10.41383166	15.96218	6.807551
UIR	7.848796251	33.47469	0.801304
UIT	11.20592124	27.25062	2.533275
Gonion	9.791211823	24.37635	2.592507
Menton	10.8856718	26.33382	2.200693
Gnathion	11.0091156	23.2225	1.102983
Pogonion	11.15456644	24.53216	0.630546
Bpoint	12.23665536	24.82242	4.218548
PNS	6.823141607	11.61833	1.993311
ANS	11.4851164	32.34936	3.874117
Apoint	8.042472715	21.40456	2.431977
LIR	12.17510456	27.02329	2.087888
LIT	6.87516	13.46533	2.834347

Table 5.5: Landmark errors across the tested images obtained by the application of grey profile matching approach for more précised localization, through permitting the interactive modification of the detected position of reference landmarks by user

Figure below shows the automatic localization of the craniofacial structure and the automatic localization of the land marks.

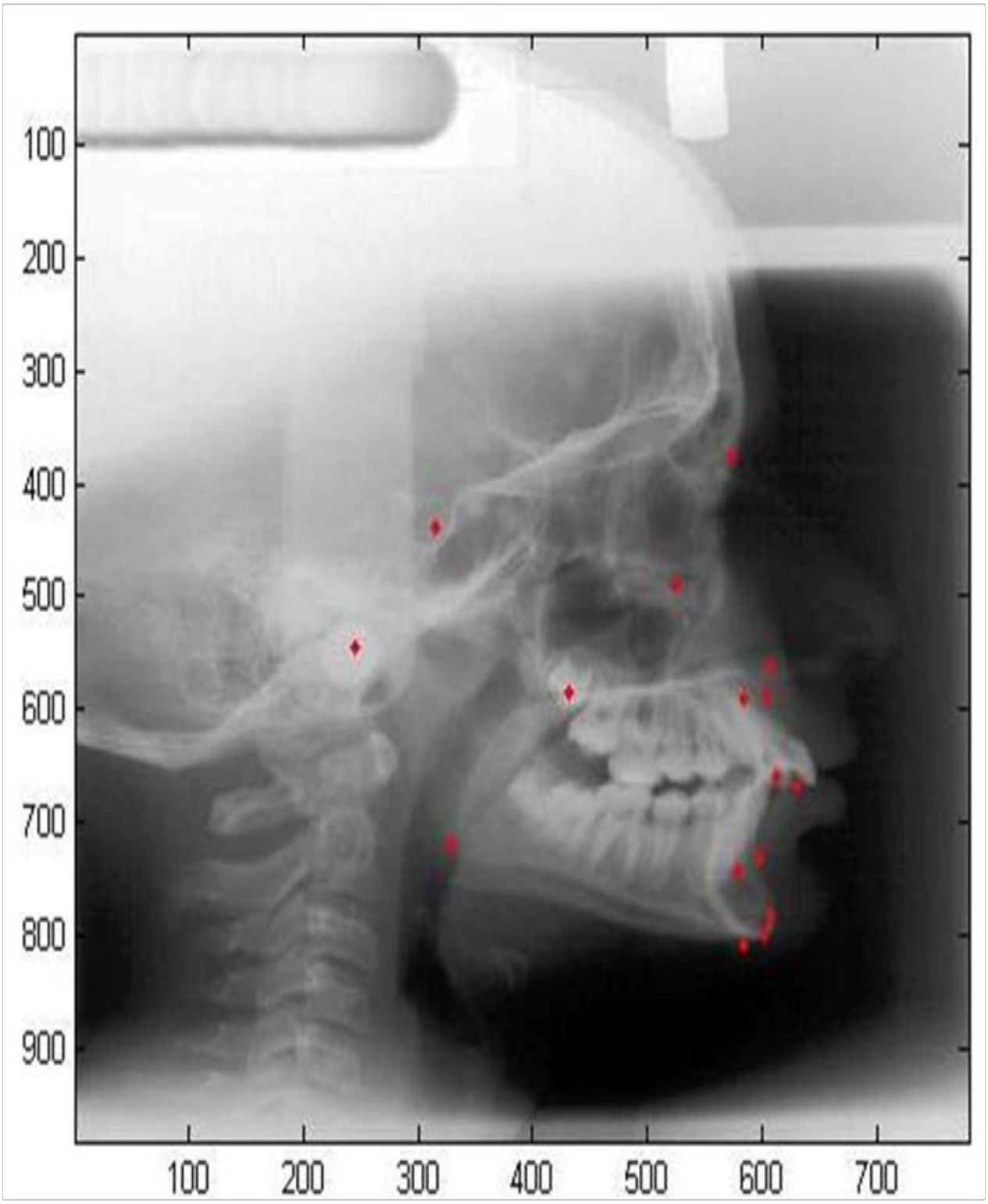


Figure 5.14: Result of localisation of landmarks on cephalometric image

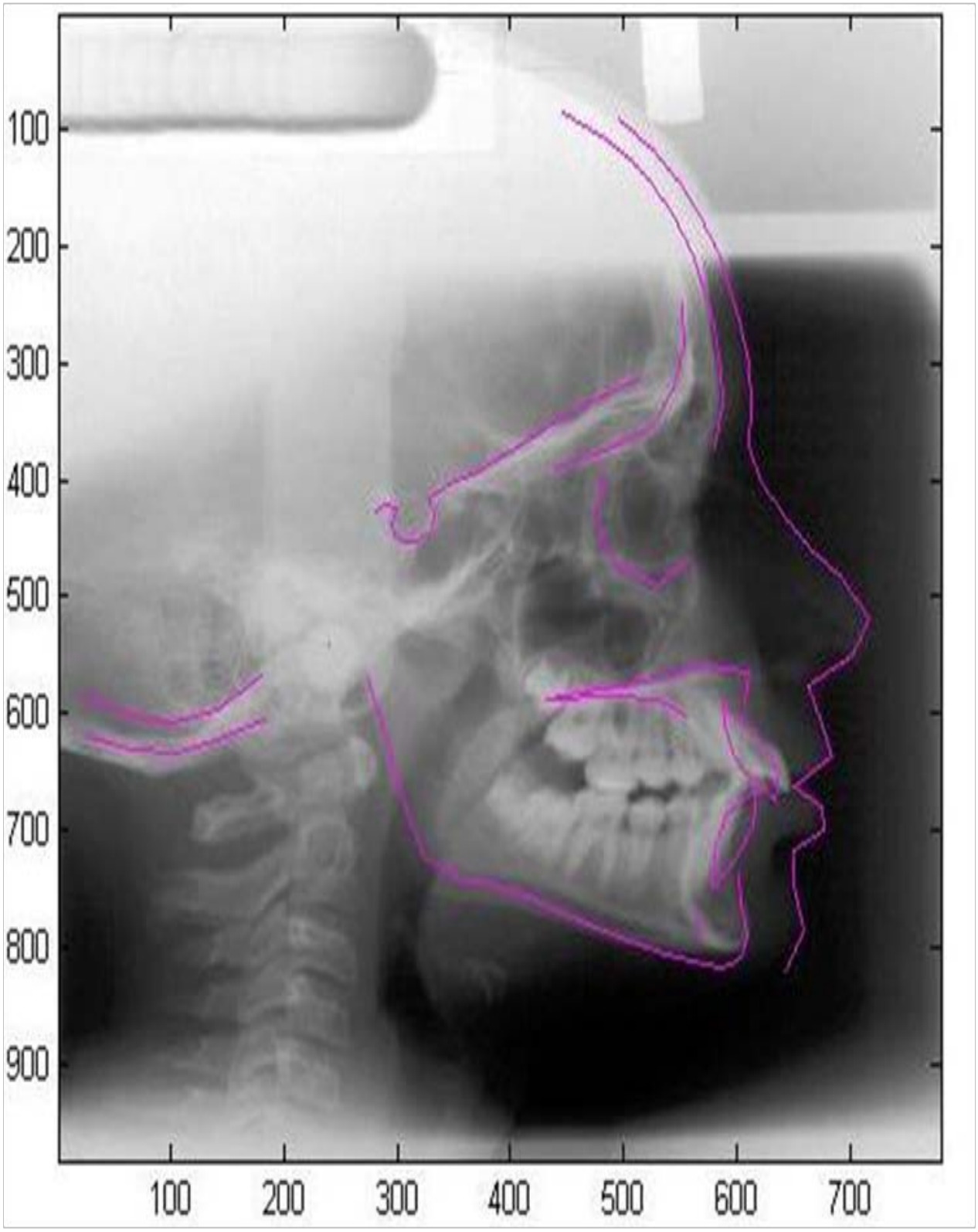


Figure 5.15: Result of localisation of craniofacial structure on cephalometric image

CHAPTER VI

6.1 Discussion : For making of more film free hospital means that automatic cephalometric land marking will become common place and more land marking algorithms will be produced for making the detection result of localization more accurate and more speedy. The fist approach is a new approach to detect the craniofacial structure in cephalometric analysis, although the second approach is previously attempted by some researchers mentioned above but there is a lot of scope to improve the result (some of this kind of thoughts are described in future work section).

The ASM approach is totally different form the method previously described to automate cephalometric analysis. The local search is employed if the template is to converge on the correct structure. The method suggested by Cootes et al.(1995) was adopted, while a statistical model of the grey level along a profile at right angles to the template at each point was derived from the training set images. The statistical match between the grey level model for each point and that found on the image yields the magnitude of each pull vector. Some parameters need to be well chosen for the search to be optimal. One is the size of the grey level model that are used to define the grey level model which is used to define the search at each point. The second parameter is that how far from each template point the search should extend. The best approach is to start with a large search distance then reduce it as the fitting converged.

sixty- three cephalometric images were used to build the model of variation in this study. For that reason a larger training set would yield a more reliable statistical model encompassing more of the natural variation. The ASM approach provides a framework for improvement as the local sub-image matching could be used to pinpoint the location of each landmark after the first level search is converged.

Until a better match is found it is better to use a smaller no. of modes, introducing more later to refine the fit. The rate at which this should be happen would ideally depend on how well the fitting has converged, as with the search distance.

This study shown that the active shape model algorithm is not sufficiently accurate to be used to automate the cephalometric analysis- this would require that the majority of landmarks to be acceptably placed in the in most images .

For the case of the first approach, As illustrated that the method is a really novel approach for finding out the craniofacial structure from the image and it is based on the true edge detection of the image, so it is less error prone, although the main problem is in the localisation of the desired structure in the image, which is an conventional problem with the template matching approach in some low quality of medical images, subsequently the remaining result are also affected. As in template matching approach we find out the correlation at each pixel to find out the best place, it takes more time if the image size is big. The proposed method for edge linking and Canny's modification is an added ornament to the algorithm. The best efficacy of this method is where the edge tracking algorithm fails at the broken edge locations. As the edge tracking algorithm is merely depends upon the manually defined length of the desired curve, the algorithm grudgingly detects other undesirable edges of length greater than or equal to the defined length from the ROI, but this detected edges can be easily ignored during visual discernment by the doctor, so as observed from the experimental result that it is somewhat dependent on image parameters, it is also observed from the experimental result that all of the desired structure can't be detected by this proposed methodology, as those structures like "Anterior Clinoid Process", "Nasal Floor, Anterior alveolar process, palatal cortical bone", "Orbit" are located at very ambiguous areas, so it is tough to locate those particular structures. The structure named "Maxillary & Mandibular central and maxillary & Mandibular first molar" is a kind of predefined model, which is manually fitted by the skilled doctors at its plausible location, so to locate this particular structure some model based approach like active shape model or active appearance model can be attempted. It is worth to be mentioned that, as the required structure of nasion curve is look alike the plough, as consulted by doctor that detection of one arm of this curve is sufficient for cephalometric analysis. By taking into account two set of images with variation in image parameters, it is demonstrated how the results of the proposed algorithm depends on the image parameters.

6.2 Future work:

Though out the experiment we can realize that human interaction is important factor . So we want to allow orthodontists to modify the intermediate result of the reference landmarks so that more accurate and good result can be obtained. Besides that we should allow user to be flexible to modify the result by three different way . In point based modification only the selected point will

be moved with the cursor. The second one is curve based mode , the user can modify the curve by selecting or moving any point on it . This will be useful for adjusting the soft tissue and hard tissue contours, i.e. they will be allowed to move and rotate each structure. It will be useful for modifying the position of the relatively anatomical structure such as molar.

One addition could be the Active appearance model (Cootes et.al 1998). This approach uses a grey level model of the entire image to drive the template matching and should make the search more robust.

The template matching approach can be improved by applying the ring projection transformation approach [42] in addition with the approach described above for rotation and scaling invariance pattern matching.

Kafieh et al. [21] described a new addition to the conventional template matching procedure by applying non linear diffusion and Susan edge detector after the result of convergence of the template matching approach described in section (4.2), This approach can be applied after the convergence of result of the approach described in [42]. It may show more better result than the result described in [21].

Four-point subdivision scheme [43], being an efficient way to construct a smooth curve interpolating an initial set of points, may be utilized to trace the craniofacial structures. The located feature points are considered to be the initial point set and connected by line segments according to the prior knowledge of structures. The scheme works by repeated refinement of the point set $\{P_i^0\}$,

$$P_{2i}^{j+1} = P_i^j$$

$$P_{2i+1}^{j+1} = \left(\frac{1}{2} + w\right)\left(P_i^j + P_{i+1}^j\right) - w(P_{i-1}^k + P_{i+2}^j)$$

where j is the subdivision level and w is a weight factor. This method keeps all original points and join new ones among them. When w is 1/16, a C^1 continuity smooth curve can be obtained in the limit.

The edge detection module proposed in first algorithmic approach can be improved by the subdividing approach discussed in second algorithmic approach.

6.3 Computational Issue: All algorithms were implemented using MATLAB 7.6 on INTEL XEON processor with 4 GB of RAM. In the full automatic mode, the average time cost of the feature point localization and structure tracing by the second approach on the image was measured to be 90 second. In addition with the human modification of the result, none case needed more than 5 min to be accomplished in the experiment.

In full automatic mode, the average time cost of the first approach to localize the craniofacial structure on the cephalometric images was measured to be merely 190 seconds. The time cost is significantly shorter than that of human operation.

6.4 Conclusion : The algorithm for the automated detection of craniofacial structures and landmarks for two-dimensional cephalometric analysis on X-ray images is proposed here. This approach comprise of image processing techniques and pattern matching for locating the region of interest so that the desired craniofacial shape can be partitioned to some structure-based regions. The experimental results show the advantage and reliability of this method. Further tests will be carried out when more suitable data are available. Such tests could contribute to further improvements on the methodology, so that more accurate result can be obtained and some approach for detection of the structures which the proposed approach are failed to detect can be contributed, that may results in more robust and more accurate localization that eventually can be accepted for the clinical purposes.

APPENDIX A: ALIGNING A PAIR OF SHAPES

If we have two shapes then we can align shape x_2 to shape x_1 by properly scaling (s), rotate (θ) and translate (t_x, t_y) the shape A to map onto shapes B. Simply mapping A onto $M(A)$ so as to minimise the following weighted sum:

$$E = (x_1 - M(s,\theta)[x_2] - t)^T W (x_1 - M(s,\theta)[x_2] - t)$$

Where $$M(s,\theta)\begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s\cos\theta)x_{jk} - (s\sin\theta)y_{jk} \\ (s\sin\theta)x_{jk} - (s\cos\theta)y_{jk} \end{pmatrix}$$

$$t = (t_x t_y \ldots\ldots\ldots\ldots\ldots\ldots t_x t_y)^T$$

Where W is diagonal matrix of weights for each point if we write:

$$a_x = s\cos\theta\,, a_y = s\sin\theta$$

A least square approach leads to a set of four linear equations, (By differentiating with respect to four of each variable a_x, a_y, t_x, t_y we got it)

$$\begin{pmatrix} X_2 & -Y_2 & W & 0 \\ Y_2 & X_2 & 0 & W \\ Z & 0 & X_2 & Y_2 \\ 0 & Z & -Y_2 & X_2 \end{pmatrix}\begin{pmatrix} a_x \\ a_y \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} X_1 \\ Y_1 \\ C_1 \\ C_2 \end{pmatrix},$$

$$X_i = \sum_{k=0}^{n-1} w_k x_{ik} \quad Y_i = \sum_{k=0}^{n-1} w_k y_{ik}$$

$$Z = \sum_{k=0}^{n=1} w_k\left(x_{2k}^2 + y_{2k}^2\right), W = \sum_{k=0}^{n-1} w_k$$

$$C_1 = \sum_{k=0}^{n-1} w_k (x_{lk}x_{2k} + y_{lk}y_{2k})$$

$$C_2 = \sum_{k=0}^{n-1} w_k (y_{lk}x_{2k} + x_{lk}y_{2k})$$

This can be solved for a_x, a_y, t_x, t_y using standard matrix methods.

APPENDIX B: PRINCIPAL COMPONENT ANALYSIS

In this section we describe principal component analysis (PCA) which can be derived by means of simple linear algebra and used for modeling shape variation.

Suppose we have N shapes and each shape is represented by n points, then each shape can be represented by: $x = [x_1, x_2 \dots \dots \dots \dots \dots \dots \dots . x_{n-1}, x_n]^T$

Consider a set covering a certain class of shapes; we will always observe certain degree of correlation. If it does not happen then there might occur two cases:

- The set either contains no variation.
- The points are purely random i.e. the points are not landmarks.

  In our case we will seek the linear combination of our dataType equation here.

$$Y = Mx$$

  First consider the mean shape $\bar{x} = \frac{1}{N}\sum_{i=1}^{n} x_i$ and the shape co-variance matrix

$$\sum_x = \frac{1}{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

  The mean of the y variables can then be expressed as:

$$y = \frac{1}{N}\sum_{i=1}^{N} y_i = \frac{1}{N}\sum_{i=1}^{N} M_{x_i} = M\bar{x}$$

  And consequently the covariance of the y's:

$$\sum_y = \frac{1}{N}(y_i - \bar{y})(y_i - \bar{y})^T$$

$$= \frac{1}{N}\sum_{i=1}^{N}(Mx_i - M\bar{x})\,(Mx_i - M\bar{x})^T$$

$$= \frac{1}{N}\sum_{i=1}^{N} M(x_i - M\bar{x})\,(M(x_i - \bar{x}))^T$$

$$= \frac{1}{N}\sum_{i=1}^{N}(x_i - M\bar{x})\,(x_i - \bar{x})^T M^T$$

$$= M\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - M\bar{x})\,(x_i - \bar{x})^T\right)M^T$$

$$= M\sum_x M^T$$

  Then if we limit ourselves to the orthogonal transformation (i.e. $M^{-1} = M^T$), left multiplication in (16) yields.

$$M^T\sum_x = \sum_y M^T$$

  Then by substituting M^T by φ gives

$$\sum_x \varphi = \varphi \sum_y$$

From (18) it can be seen that φ is chosen as the (column) Eigen vectors of the symmetric matrix $\sum_x$, then the co-variance of the transformed shapes, $\sum_y$, becomes a diagonal matrix of the Eigen values. In the case of correlated points the smallest Eigen values will be (close to) zero and the corresponding Eigen vector could be omitted from φ which reduces the length of y .

In order to back transform from the new set of variables, y, we invert (13), remembering that M is orthogonal. This establishes a linear transformation that de-correlate the data vectors, the transformation matrix must be Eigen vectors of the covariance matrix of the original data .

$$x = M^{-1}y = M^{T}y = \varphi\, y$$

Typically one could apply PCA on variables with zero mean (notice that the φ is unchanged).

$$y = M(x - \bar{x}) , \qquad x = \bar{x} + \varphi x$$

This celebrated method of dealing with redundancy in multivariate data known as principal component analysis(PCA).

It should be mentioned that the PCA could be derived as a variance maximizing change of basis given an orthonormality constraint using the Lagrange formulation.

REFERENCES:

[1] R. Moyers (1988) Planning Orthodontic treatment in: Handbook of orthodontics. Year Book Medical Pub. 4th ed.

[2] C. Szuhanek (2006) Periodontal Implications in Orthodontics PhD Thesis ”Victor Babes”UMPh Timisoara

[3] F. Glavan, E. Bratu (2005) Ortodontie si ortopedie dento-faciala. Ed.Mirton, Timisoara

[4] I. El-Feghi, M.A. Sid-Ahmed and M. Ahmadi, “Automatic Localization of Craniofacial Landmarks for Assisted Cephalometry, ” Patt. Recog., vol. 37, no.2, pp. 609-621, 2004.

[5] A. Levy-Mandel, A. Venetsanopoulos, and J. Tsotsos, “Knowledgebased landmarking of cephalograms,” Comput. Biomed. Res., vol. 19,no. 3, pp. 282–309, 1986.

[6] S. Parthasarathy, S. T. Nugent, P. G. Gregson, and D. F. Fay, “Automatic landmarking of cephalograms,” Comput. Biomed. Res., vol. 22,no. 3, pp. 248–269, 1989.

[7] D. N. Davis and C. J. Taylor, “Ablackboard architecture for automating cephalometric analysis,” Med. Inf., vol. 16, no. 2, pp. 137–149, 1991.

[8] W. Tong, S. T. Nugent, P. G. Gregson, G. M. Jensen, and D. F.Fay, “Landmarking of cephalogams using a microcomputer system,”Comput. Biomed. Res., vol. 23, no. 4, pp. 358–397, 1990.

[6] J. C. Ren, D. Liu, and J. L. Shao et al., “A knowledge-based automaticcephalometric analysis method,” in Proc. 20th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Soc., 1998, vol. 20, no. 2, pp. 723–727.

[7] M.Wei, Z. Zhou, and W. Ding et al., “Prelimary study of computerized automatic identification of landmarks in X-ray cephalometry,” Chin. J. Oral Maxillofac. Surg., vol. 10, no. 4, pp. 283–287, 2000.

[8] D. J. Rudolph, P. M. Sinclair, and J. M. Coggins, “Automatic computerized radiographic identification of cephalometric landmarks,” Am. J. Orthod. Dentofac. Orthop., vol. 113, no. 2, pp. 173–179, 1998.

[9] J. Cardillo and M. A. Sid-Ahmed, “An image processing system for locating craniofacial landmarks,” IEEE Trans. Med. Imag., vol. 13, no.2, pp. 275–289, Jun. 1994.

[10] Y. T. Chen, K. S. Cheng, and J. K. Liu, “Improving cephalogram analysis through feature subimage extraction,” IEEE Eng. Med. Biol. Mag., vol. 18, no. 1, pp. 25–31, Jan.-Feb. 1999.

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models: their training and application,” Comput. Vis. Image Understanding, vol. 61, no. 1, pp. 38–59, 1995.

[12] T. J. Hutton, S. Cunningham, and P. Hamrnond, “An evaluation of active shape models for the automatic identification of cephalometric landmarks,” Eur. J. Orthodont., vol. 22, no. 5, pp. 499–508, 2000.

[13] B. Romaniuk, M. Desvignes, M. Revenu, and M. J. Deshayes, “Linear and non-linear model for statistical localization of landmarks,” in Proc. 16th Int. Conf. Pattern Recognit., 2002, vol. 4, pp. 393–396.

[14] ——, “Shape variability and spatial relationships modeling in statistical pattern recognition,” Pattern Recognit. Lett., vol. 25, no. 2, pp. 239–247, 2004.

[15] I. El-Feghi, M. A. Sid-Ahmed, and M. Ahmadi, “Automatic localization of craniofacial landmarks for assisted cephalometry,” in Proc. Int. Symp. Circuits and Systems, 2003, vol. 3, pp. 630–633.

[16] ——, “Automatic localization of craniofacial landmarks for assisted cephalometry,” Pattern Recognit., vol. 37, no. 3, pp. 609–621, 2004.

[17] S. Chakrabartty, M. Yagi, T. Shibata, and G. Gauwenberghs, “Robust cephalometric landmark identification using support vector machines,” in Proc. 2003 Int. Conf. Multimedia and Expo, 2003, vol. 3, pp. 429–432.

[18] B. Romaniuk, M. Desvignes, M. Revenu, and M. J. Deshayes, Feb. 2005, Contour Tracking by Minimal Cost Path Approach. Application to Cephalometry [Online]. Available: http://www.lis.inpg.fr/ pages_perso/desvignes/publi/icip04.pdf.

[19] Weining Yue*, Dali Yin, Chengjun Li, Guoping Wang, Member, IEEE, and Tianmin Xu, Automated 2-D Cephalometric Analysis on X-ray Images by a Model-Based Approach, ieee transactions on biomedical engineering, vol. 53, no. 8, august 2006.

[20] Lihong Ma,Shengmin Jiang Yu Zhang, Chunyi Lin Hanqing Lu , Craniofacial Landmark Detection by Layered Diffusion and Dilated Skeleton Maps, 1-4244-0342-1/06/$20.00 ©2006 IEEE.

[21] Rahele Kafieh, Alireza mehri,Saeed sadri, Automatic Landmark Detection in Cephalometry Using a Modified Active Shape Model with Sub Image Matching, 978-1-4244-1625-7/07/$25.00 ©2007 IEEE.

[22] I.El-Feghi, M. Galhood, M. Sid-Ahmed, M. Ahmadi Automated 2-D Cephalometric Analysis of X-ray by Image Registration Approach based on Least Square Approximator 30th Annual International IEEE EMBS Conference Vancouver, British Columbia, Canada, August 20-24, 2008.

[23] K. Tanaka, M. Sano, S. Ohara , and M. Okudira. “A parametric template method and its application to robust matching,” Proc. of IEEE Conf. on Com-puter Vision and Pattern Recognition, pp. 620-627, 2000.

[24] P. Burt, and E.H. Adelson, “The Laplacian Pyramid as a Compact Image Code,” IEEE Trans. Comm., vol. 9, no. 4, pp. 532-540, 1983.D. Marr and E. Hildreth, “Theory of Edge Detection,” Proc. RoyalSoc. London, B-207, pp. 187-217, 1980.

[25] D. Marr and E. Hildreth, “Theory of Edge Detection,” Proc. RoyalSoc. London, B-207, pp. 187-217, 1980.

[26] A. Witkin, “Scale-Space Filtering,” Int’l Joint Conf. Artificial Intelligence, pp. 1,019-1,021, Karlsruhe, West Germany, 1983.

[27] Jamy Li, Jonathan Randall, Ling Guan, PERCEPTUAL IMAGE PROCESSING FOR DIGITAL EDGE LINKING, CCECE 2003 - CCGEI 2003, Montri al, May/mai 2003 0-7803-7781-8/03/$17.00 8 2003 IEEE.

[28] Samuel H. Chang, Leiguang Gong2, Maoqing Li, Xiaoying Hu and Jingwen Yan Small Retinal Vessel Extraction Using Modified Canny Edge Detection. 978-1-4244-1724-7/08/$25.00 ©2008 IEEE.

[29] J. Canny, “A computational approach to edge detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. , pp. 679–698, 1986.

[30] I. El-Feghi,M.A. Sid-Ahmed,M. Ahmadi , Automatic localization of craniofacial landmarks for assisted cephalometry, 2003 Pattern Recognition Society. Published by Elsevier Ltd.

[31] S.L. Chiu, Fuzzy model identi8cation based on cluster estimation, J. Intell. Comput. Fuzzy Systems 2 (3)(1994) 267–278

[32] P.R. Krishna, L.N. Kanal, Classi8cation, Pattern Recognition,and Reduction of Dimensionality, in: Handbook of Statistics,Vol. 2, North-Holland, Amsterdam, 1982.

[33] J. Makhoul, S. Roucos, H. Gish, Vector quantization in speech coding, Proc. IEEE 73 (11) (1985) 1551–1588.

[34] C. Goodal, "Procrustes methods in the statistical analysis of shape," J.Roy. Statist. Soc. Ser. B, vol. 53, no. 2, pp. 285–339, 1991.F. L. Bookstein, "Landmark methods for forms without landmarks: localizing group differences in outline shape," Med. Image Anal., vol. 1,no. 3, pp. 225–244, 1997

[35] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," J. Opt. Soc. Am. A, vol. 4, no. 4, pp. 629–642, 1987.

[36] F. L. Bookstein, "Landmark methods for forms without landmarks: localizing group differences in outline shape," Med. Image Anal., vol. 1,no. 3, pp. 225–244, 1997.

[37] R.A. Johnson & D.W. Wichern : Multivariate Statistics, a Practical Approach, Chapman & Hall 1988.

[38] D.R.Bailes, C.J.Taylor: T he Use of Symmetry Chords for Expressing Grey Level Constraints. in Proc. British Machine Vision Conference. Springer-Verlag, 1992, pp.296-305.

[39] M. Kass, A. Witkin and D. Terzopoulos : Snakes: Active Contour Models. In: Proc. First International Conference on Computer Vision, pp 259-268 IEEE Computer Society Press, 1987.

[40] T.F.Cootes, C.J.Taylor, D.H.Cooper and J.Graham: Training Models of Shape from Sets Examples. In: Proc. British Machine Vision Conference. Springer-Verlag, 1992, pp.9-18.

[41] P.J Burt , The Pyramid as a structure for efficient computation, in multi resolution image processing and analysis, Ed,Rosenfield,pub,Springer-Verlag, 1984,pp,6-37

[42] Translation, Rotation,and Scale-Invariant Template Matching by Ti-Hsein,Chen-Hsing Chen.

[43] N. Dyn, D. Levin, and J. Gregory, “A 4-point interpolatory subdivision scheme for curve design,” Comput. Aided Geom. Des., vol. 4, pp.257–268, 1988