

# IMAGE PROCESSING FOR PLASTIC SURGERY PLANNING

by  
LEILA FAVAEDI  
*MSc*

A Thesis submitted in fulfilment of requirements for the degree of  
Doctor of Philosophy of Imperial College London and  
Diploma of Imperial College

Communications and Signal Processing  
Department of Electrical and Electronic Engineering  
Imperial College London  
2010

*To my parents with heartfelt love and gratitude*

# Abstract

This thesis presents some image processing tools for plastic surgery planning. In particular, it presents a novel method that combines local and global context in a probabilistic relaxation framework to identify cephalometric landmarks used in Maxillofacial plastic surgery. It also uses a method that utilises global and local symmetry to identify abnormalities in CT frontal images of the human body. The proposed methodologies are evaluated with the help of several clinical data supplied by collaborating plastic surgeons.

# Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

First, I would like to thank Professor Maria Petrou who has been a most exceptional supervisor. I have benefited from her thoughtful guidance and constant encouragement, for which I am extremely grateful.

I am also grateful to Plastic Surgeon Professor Peter Haers from the Royal Surrey Hospital who supplied me with data used in the first part of the thesis. The data used in the second part of the thesis as well as the expert advice were kindly supplied by Plastic Surgeon Professor Niall Kirkpatrick of Hammersmith Hospital, the contribution of whom is gratefully acknowledged.

I would also like to thank all my colleagues at EEE Department, especially my friends at room 805a and Ms. Glenys Benson.

Finally, I would like to thank my parents and sisters for their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

# Contents

<b>Abstract</b>	<b>3</b>
<b>Acknowledgments</b>	<b>4</b>
<b>Contents</b>	<b>5</b>
<b>Abbreviations</b>	<b>8</b>
<b>Chapter 1. Introduction</b>	<b>9</b>
1.1 Introduction and Motivation . . . . .	9
1.2 Problem Statement . . . . .	11
1.2.1 Cephalometric Landmark Identification . . . . .	11
1.2.2 Asymmetry Identification Using Generalised Symmetry Axis . . . . .	12
1.3 Statement of Originality . . . . .	13
1.4 Structure of the Thesis . . . . .	13
1.5 Publications from the Thesis . . . . .	14
<b>Chapter 2. Image Pre-processing</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Data Pre-processing . . . . .	16
2.2.1 Creating a mask . . . . .	16
2.2.2 Bone area enhanced by histogram equalization . . . . .	18
2.3 Active Contours . . . . .	18
2.4 Parametric Active Contours . . . . .	20
2.4.1 Theory . . . . .	20
2.4.2 Practical application . . . . .	26
2.4.3 Experiments . . . . .	34
2.5 Geometric active contours . . . . .	35
2.5.1 Theory . . . . .	38
2.5.2 Practical Implementation . . . . .	43
2.5.3 Experiments . . . . .	45

2.5.4	Discussion and conclusions . . . . .	48
2.6	Conclusions from this chapter . . . . .	48

### **Chapter 3. Cephalometric Landmarks Identification Using Probabilistic**

<b>Relaxation</b>		<b>49</b>
3.1	Introduction . . . . .	49
3.2	Landmark Errors . . . . .	50
3.3	Previous Work . . . . .	50
3.3.1	Prior Knowledge-Based methods . . . . .	51
3.3.2	Learning . . . . .	52
3.4	Active Shape Models (ASMs) . . . . .	54
3.5	Theoretical Background . . . . .	55
3.6	The Proposed Method . . . . .	61
3.6.1	Data Preprocessing . . . . .	61
3.6.2	A DFT-based technique for scaling measurement . . . . .	61
3.6.3	Matching Procedure . . . . .	71
3.6.4	Discussion and conclusions . . . . .	85
3.7	Conclusions from this chapter . . . . .	87

### **Chapter 4. Asymmetry Identification Using Generalised Symmetry Axis** 88

4.1	Introduction . . . . .	88
4.2	Symmetry . . . . .	89
4.3	Types of symmetry . . . . .	89
4.4	Previous work on symmetry detection . . . . .	90
4.4.1	Symmetry detection from borders . . . . .	91
4.4.2	Symmetry detection from image intensity . . . . .	93
4.5	The proposed method . . . . .	94
4.5.1	Dividing the image into two grossly symmetric parts . . . . .	95
4.5.2	Registration . . . . .	95
4.5.3	Refinement Stage . . . . .	98
4.5.4	Symmetry point identification . . . . .	109
4.5.5	Multiple symmetry points . . . . .	109
4.5.6	Results . . . . .	110
4.5.7	Improving the symmetry axis by using a lateral continuity constraint	119
4.5.8	Straightening the symmetry axis and identifying the abnormalities	130
4.6	Conclusions from this chapter . . . . .	132

### **Chapter 5. Evaluation** 133

5.1	Introduction . . . . .	133
-----	------------------------	-----

---

5.2	Results . . . . .	133
5.3	Discussion and Conclusions . . . . .	137
<b>Chapter 6. Conclusions and Future Work</b>		<b>146</b>
6.1	Cephalometric Landmark Identification . . . . .	146
6.2	Asymmetry Identification Using Generalised Symmetry Axis . . . . .	147
<b>Bibliography</b>		<b>149</b>
<b>Appendix A. The Cephalometric Landmarks And Their Meaning</b>		<b>163</b>
<b>Appendix B. Algorithm for Local Histogram Equalization</b>		<b>165</b>

# Abbreviations

**DFT:** Discrete Fourier Transform

**IDFT:** Inverse Discrete Fourier Transform

**LPM:** Log-Polar Mapping

# Chapter 1

## Introduction

### 1.1 Introduction and Motivation

**T**HE first medical image that showed the inside of the human body without having to cut into the flesh was created after the discovery of X-rays by Roentgen in 1895. Although the discovery of the X-rays was a remarkable step forward in the history of medicine, for decades the only imaging modality available was projection radiography which only used film to create X-ray images. It was after the invention of the diagnostic technique of X-ray computed tomography (CT) by Hounsfield and Cormack in 1970s [1], that medical imaging has made significant progress. Since then, new digital imaging methods have been invented to reveal different characteristics of the body (e.g., magnetic resonance imaging (MRI), ultrasound and positron emission tomography) and even projection radiography has become digitized. Introducing digital imaging into radiology has allowed digital image processing to be widely used in medical applications and provide many benefits for various areas of the medical profession. From diagnosis to treatment, imaging technologies have fundamental importance in medical fields such as surgery, laboratory medicine, nuclear medicine, radiation therapy, or diagnostic radiology. The imaging technologies used in medical applications can be divided into two groups, depending on whether the resulting data give a representation of the body's internal anatomical structures, or a mapping of physiological functions. The main potential aspect of image processing is to help surgeons

for as fast diagnosis and treatment as possible [2] [3] [4].

There have been various research efforts in the area of medical image analysis over the past 30 years. The first papers about computer analysis of radiographic images were written by Becker et al. [5] and Myers et al. [6] in 1964. However, as it was noted by Duncan et al. [7] and Ginneken et al. [8], it was in the 1970s when a few researchers started treating medical image analysis as a unique information processing problem. Duncan and Ayache divided the research efforts that are mainly based on pattern recognition, image processing and computer vision into four time periods (pre-1980-1984; 1985-1991; 1992-1998; 1999 and beyond) [7]. They called the first period as the era of 2D image analysis. The works in this era mainly focused on segmenting structures, registering two images and quantifying cardiac motion from image sequences, problems that are still being addressed. The automatic detection of tumors in chest X-ray images is an example of the early works in 1973 [9] [10]. In terms of image registration, the main application area that researchers worked on was digital subtraction angiography in the early 1980s [11] [12]. Interestingly, in this era, some researchers were truly ahead of their time by developing ideas that become useful in the current years, such as a fully 3D edge detector [13], 3D deformable object segmentation [14] and elastic matching of atlases to image data [15]. The second period was the time when knowledge-based strategies came to the forefront and much research was based on them (e.g. [16] [17] [18] [19]). Also, it was in the second period that the first efforts at fully automatic computer-aided diagnosis (CAD) of X-ray mammograms were reported [20]. In addition, there was much research on the classic problems of identifying boundaries [21] [22] [23] and 2D image matching [24] [25]. From 1992 to 1998, medical image analysis came into a new stage as fully 3D images became a key goal in developing methodologies. Medical imaging modalities improved significantly during this time. However, these improvements caused researchers to face and handle more and more complex problems. During this time period, image segmentation and image registration continued to be important and challenging areas of research [26] [27] [28] [29] [30] [31]. Also, the improving capabilities of medical image acquisition, computers, and computer graphics technology helped medical image analysis researchers into areas of interactive analysis and image-guided intervention. Researchers needed to face a range

of new challenges such as the problems of identifying structure from both preoperative and intraoperative information in near real time, and the registration of different types of intraoperative information to preoperative images (image-based atlases). An example of a clinically successful line of research is the use of image guidance for hip replacement surgery [32] [33].

Since the start of the new century, more advanced imaging and computing technologies have facilitated work in image guided procedures, but medical image analysis has continued to be an active area of research, with many encouraging results, but also with a number of difficult problems still to be addressed. These may include enormous image data base processing, automatic detection of pathologic cases by enhancement of details and recognition of patterns, accurate measurements of the distortions in the medical images, and prediction of results to allow planning of treatment. Therefore, the importance of utilization of image processing capabilities in the medical field, would encourage any engineer to work in this field.

## 1.2 Problem Statement

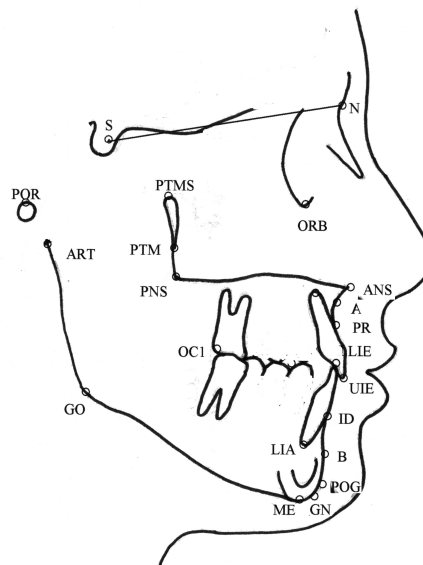
The purpose of this thesis is to help plastic surgery planning in two different fields, namely in the automatic identification of cephalometric landmarks and in the identification of asymmetries between the left and right parts of the human body.

### 1.2.1 Cephalometric Landmark Identification

Cephalometry is the scientific measurement of the human head which was first introduced by Broadbent in 1931 [34]. Since then it has been largely utilized to study the bony structure of the face for the diagnosis of abnormality, surgical planning and treatment, and studies on craniofacial growth. It is based on a set of agreed upon feature points (cephalometric landmarks or bony landmarks) from standardized lateral skull radiographs, or cephalograms<sup>1</sup>. The number of bony landmarks which are defined is quite

---

<sup>1</sup>A 2D X-ray image of the skull



**Figure 1.1: Cephalometric landmarks**

large. However, according to Rakosi [35] only a small number of these landmarks are frequently used. In addition, despite all the efforts to automate the process of locating the landmarks, manual locating is still used. The automatic methods are not currently used in clinical routine because of their high failure rate. Therefore, operator experience and knowledge of anatomy and radiography are vital factors. However, locating landmarks on an X-ray image by even an experienced person takes up to 20 minutes [36] [37] [38].

In this thesis, we develop methodology to locate cephalometric landmarks, supplied as a shape template shown in figure 1.1, on X-ray images, in an automatic way. The proposed method is based on probabilistic relaxation to identify the locations of these landmarks on X-ray images. See also Appendix A for a list of the landmarks shown in figure 1.1.

### 1.2.2 Asymmetry Identification Using Generalised Symmetry Axis

The vast majority of facial and head trauma patients with isolated or severe injuries are injured as a result of road traffic accidents, falling, or beating. Internal bleeding, injuries and deformities in these kinds of patient destroy the symmetry, for example, of the frontal view of the face. In this thesis, we develop methodology for identifying the breaking of

symmetry in CT images of parts of the human body, that are expected to be symmetric, and identifying the degree of asymmetry, as a diagnostic support tool for surgeons.

### 1.3 Statement of Originality

As far as the author is aware, the following aspects of the thesis are believed to be original contributions:

1. Proposed probabilistic relaxation method for identifying cephalometric landmarks.
2. Proposed method for generating the local axis of symmetry.

### 1.4 Structure of the Thesis

This thesis is laid out in six chapters.

Chapter one gives an introduction of the thesis.

Chapter two describes some image preprocessing methods that are used in this thesis. First, the description of the method to create a mask to remove background noise is given. Then, the chapter discusses active contour models which are useful in segmenting the information of interest. First, the Kass algorithm [39], which is a parametric active contour, is described in detail. Then, the Chan and Vese algorithm [40], that is a geometric active contour, is described and implemented.

Chapter three provides some literature survey in cephalometric landmark identification methods. Then, the proposed method to identify cephalometric landmarks on X-ray images is described in detail.

Chapter four gives some literature survey of previous work on symmetry detection. Then, the chapter describes the proposed algorithm for generating the local axis of symmetry in CT images.

Chapter five presents the evaluation results of axes of symmetries and identifying deformities.

Chapter six concludes with future work.

## 1.5 Publications from the Thesis

1. “Automatic Extraction of Local Axis of Bone Symmetry in CT Images”, L. Favaedi and M. Petrou, 30th Annual International Conference of the IEEE-Engineering-in-Medicine-and-Biology-Society, 2008, pp 478-481.
2. “Cephalometric Landmarks Identification Using Probabilistic Relaxation”, L. Favaedi and M. Petrou, 32nd Annual International Conference of the IEEE-Engineering-in-Medicine-and-Biology-Society, 2010.
3. “Asymmetry Identification Using Generalised Symmetry Axes”, L. Favaedi and M. Petrou, Journal of Information Technology and Multimedia.

## Chapter 2

# Image Pre-processing

### 2.1 Introduction

**I**N this thesis, we are dealing with two types of image (X-ray images and CT images) for two different purposes (see previous chapter (Introduction chapter)).

X-ray images are images which can provide patient-specific information on structure and functionality and they are normally used for the detection of pathology of the skeleton system. In facial surgical planning, it is vital to find bones from X-ray images. However, in the human skeleton, the skull (the skeleton of the head) has the most complex bony structure in the body and it comprises many bones that are closely fitted together. Therefore, finding facial bones, which are marked manually at present, is difficult and takes a lot of time.

We are also dealing with CT images in this thesis. CT images are produced by using the CT scan which has become an important adjunct to X-rays. The CT scan uses multiple X-ray beams projected at different angles in conjunction with computer resources to create 3D images. Each image reveals a different level of tissue that resembles slices. CT images have advantages over normal X-ray images in distinguishing between different types of tissue.

The important step to analyse the type of medical data we are concerned with, is to extract the bone areas. However, extraction of the bone areas from an X-ray image is more complex than from CT images, because the bone regions are not uniform in intensity and quality. For this purpose, first we need to use some data pre-processing methods on the X-ray images. Then, we are going to use active contour models which are widely used for medical image segmentation. Two methods are implemented and the shapes of contours extracted from the images by them are compared and contrasted in order to decide upon which method we shall use in each part of the thesis.

## 2.2 Data Pre-processing

To identify the cephalometric landmarks, first we are interested in enhancing the contrast in the bone region of a noisy X-ray image (see figure 2.1(a)) as much as possible. To do that, the following procedures are used.

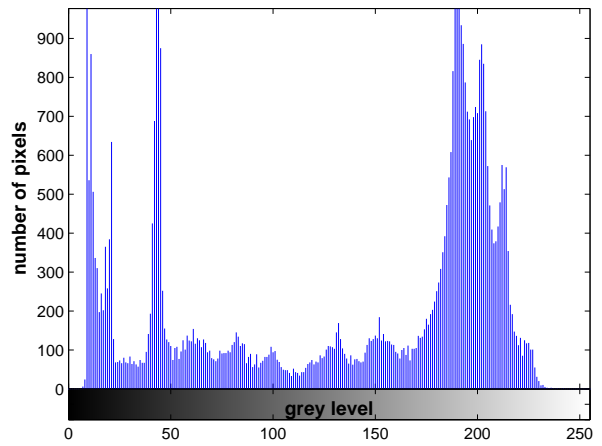
### 2.2.1 Creating a mask

To create a mask to separate the full head from the background, one has to measure the similarity between pixel values. The important role of the mask is explained in the next chapter. There are many segmentation operations that could be used. Thresholding is one of them. However, to have all the bony and soft tissue parts in one step is not possible. Thus, first we try to separate the bone area from the image.

A cephalogram represents different tissues by greyscale intensity values. The soft tissues have very low intensity values whereas bones have high intensity values. Therefore, to choose the best threshold to separate the head from the background, we must first separate the bone region of the cephalogram. For this purpose, the histogram of the image [41] [42] is first built (see figure 2.1(b)). To choose the ideal threshold value to split this image into two regions, Otsu's thresholding [43], that is often used to calculate the optimal threshold without using any a priori information about the image, is used in this project. The threshold was found to be 113. Then, to remove small extra regions and



(a)



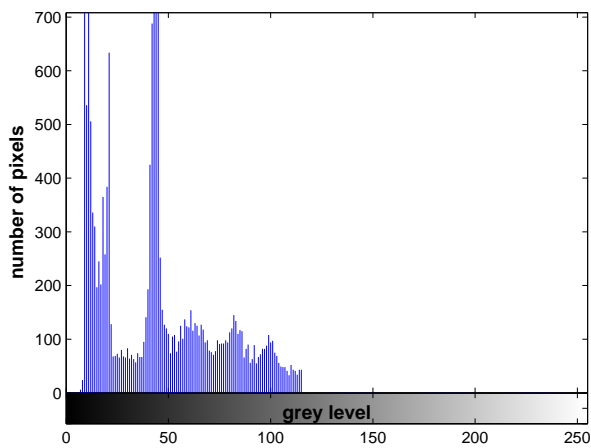
(b)



(c)



(d)



(e)



(f)

**Figure 2.1: Creating a mask.** (a) An X-ray image with size of  $223 \times 240$ . (b) The histogram of the X-ray image (a). (c) First created mask to separate the bone region. (d) The area under the mask is removed. (e) The histogram of the image (d) without considering the masked values. (f) Final mask.

holes, the segmented bone region can be processed by using mathematical morphology (a  $3 \times 3$  structuring element applied to the thresholded image). The result is shown in figure 2.1(c) where the bright object represents the bones.

By considering the result as an initial mask on the original image, the next step is to find the histogram of the area that is not under the mask (see figure 2.1(d)). The histogram is shown in figure 2.1(e) (the threshold was found to be 53). The result of applying this threshold to the original image is shown in figure 2.1(f) where the bright object represents the bone and soft tissue areas. If there are any small regions, they may also be removed. Only the area under the mask was kept for further processing and the rest of the image was disregarded.

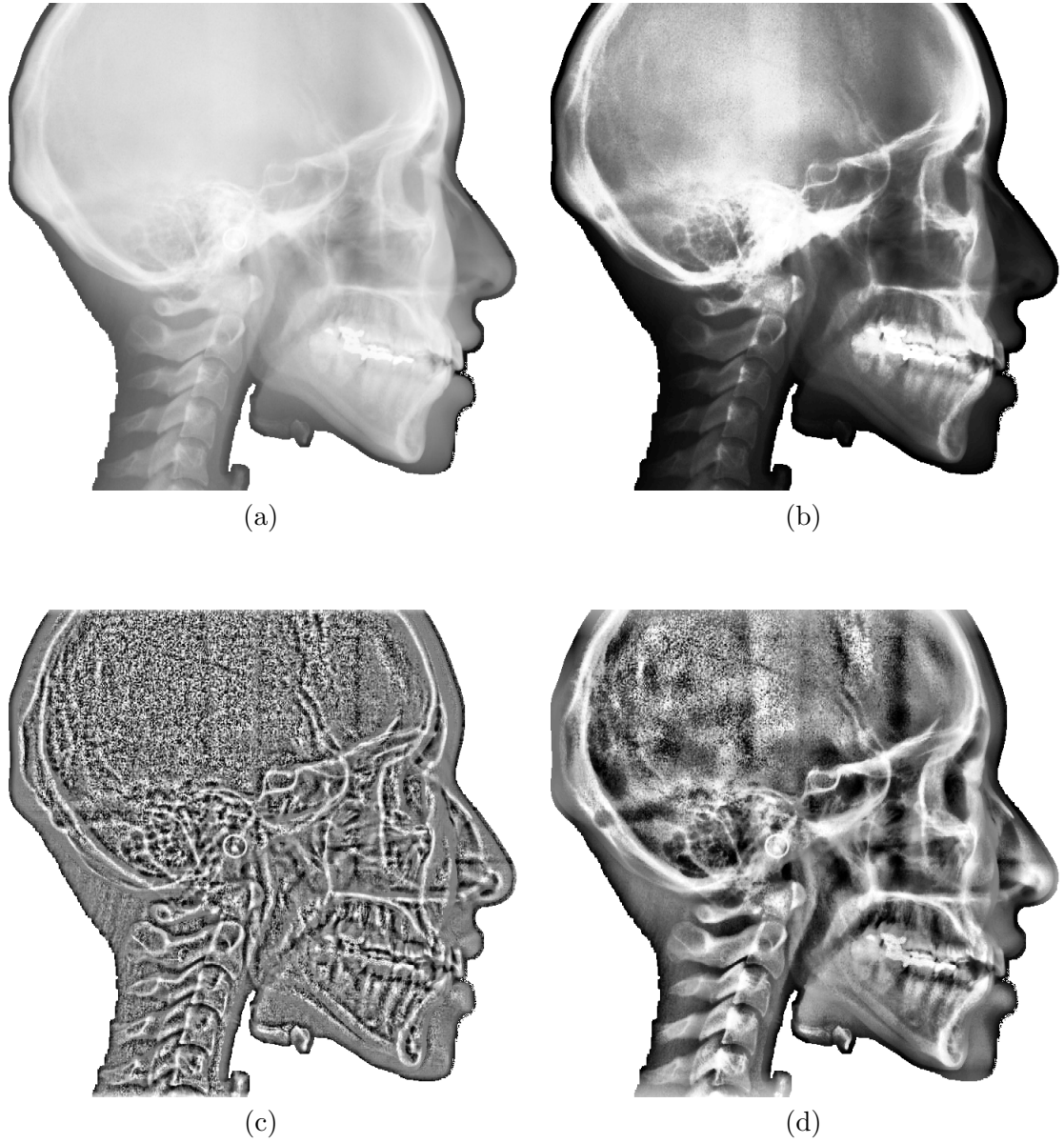
### 2.2.2 Bone area enhanced by histogram equalization

Global histogram equalization is the most widely used technique for contrast enhancement. Figure 2.2(b) shows the result of histogram equalization applied to the masked bone region of figure 2.2(a).

Global histogram equalization is a suitable technique for overall enhancement. However, global histogram equalization ignores inhomogeneities in the contrast. Local histogram equalization which uses a sliding window to equalize the histograms of local areas, is more suitable for enhancing images with inhomogeneous contrast. Figure 2.2(c) and 2.2(d) show the results of local histogram equalization with a window of size  $17 \times 17$  and  $71 \times 71$ , respectively. After the above preprocessing, the bone region has maximum internal contrast. The algorithm used to perform it is given in appendix B.

## 2.3 Active Contours

The active contour model is a deformable contour that is defined as an energy minimising spline. It was initially developed by Kass et al. in 1987 [39]. The model has been often used in computer vision and image analysis to detect objects, and to describe their shape.



**Figure 2.2:** (a) Image after using the mask on the original image. (b) Global histogram equalization. (c) Local histogram equalization with a window of size  $17 \times 17$ . (d) Local histogram equalization with a window of size  $71 \times 71$ .

An active contour is a physically-motivated model that can deform itself to locate itself in image curves or edges. There are two main types of active contour model [44] [45]:

- *Parametric model* – This model represents contours explicitly as parameterised curves [39].
- *Implicit (Geometric) model* – This model is based on the theory of curve evolution and represents curves implicitly as level sets of two-dimensional scalar functions [46] [47] [48].

## 2.4 Parametric Active Contours

The original Active Contour Model or snake was a parametric one which was proposed by Kass et al. in 1988 [39]. It can be thought of as an open or a closed curve that represents some salient shape. A general way of specifying the snake's properties is by using a quantity called energy. First an initial snake is placed on the image and then its energy is minimised. The parametric model consists of an elastic curve and can move under the influence of internal forces coming from within the curve itself and external forces computed from image data. It is also defined by a set of control points (or nodes), rather than by a function describing the overall contour. Therefore, it looks at the effect of moving nodes within the model instead of the overall energy of the snake.

### 2.4.1 Theory

The parametric active contour is defined in the  $(x, y)$  plane of an image and is represented by a set of control points in sequence (see Figure 2.3). Here,  $v(s)$  represents a control point (node) on the curve and the arc length  $s$  is a parameter. A snake is closed if  $v(0) = v(1)$ .

The snake is associated with an *energy function*  $E_{snake}$  which is defined as the sum

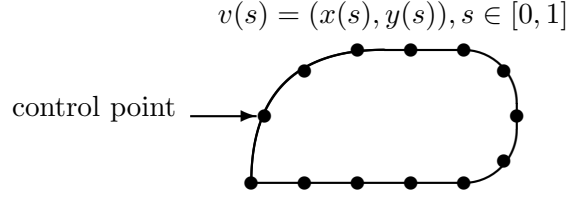


Figure 2.3: The parametric active contour

of three energy terms:

$$E_{snake}(v(s)) = E_{internal}(v(s)) + E_{image}(v(s)) + E_{constraint}(v(s)) \quad (2.1)$$

Here,  $E_{snake}(v(s))$  is the energy of node  $v(s)$ ,  $E_{internal}$  is the internal energy of the splines between a node and neighbouring nodes due to bending,  $E_{image}$  represents the image forces, and  $E_{constraint}$  is due to external constraint forces we impose on the shape.

The curve is allowed to change shape and position by minimising its own energy along its length, so that the final optimal position of the contour will have minimum energy.

### Internal Energy

The internal energy of the snake specifies the tension or smoothness of the contour and depends on the intrinsic properties of the curve. In other words, it describes the elasticity and smoothness of the contour and it is caused by stretching and bending.

The internal energy consists of two terms: the elastic term and the bending term, written as

$$E_{internal}(v(s)) = E_{elastic}(v(s)) + E_{bending}(v(s)) = \frac{1}{2} \left( \alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2 \right) \quad (2.2)$$

where,

$$v_s(s) = \frac{\partial v(s)}{\partial s}$$

$$v_{ss}(s) = \frac{\partial^2 v(s)}{\partial s^2}$$

and

- $|v|$  is the magnitude of  $v$
- $\alpha(s)$  controls tension, elasticity and stretching
- $\beta(s)$  controls stiffness, rigidity and bending

If  $\beta(s) = 0$  at a point, then the function is discontinuous in its tangent and it may develop a corner at that point. This is because in this case the second derivative of the curve is not minimised.

If  $\alpha(s) = \beta(s) = 0$  then this also allows a break in the contour, a positional discontinuity.

### Image forces

The image energy term in equation (2.1) encourages the snake to lock on the salient features of the image. It is derived from the image data and consists of three different energy functionals. These energy functionals encourage the snake to lock itself at the positions of lines, edges, and terminations. It is defined as follows

$$E_{image}(v(s)) = w_{line}E_{line}(v(s)) + w_{edge}E_{edge}(v(s)) + w_{term}E_{term}(v(s)) \quad (2.3)$$

where  $w_{line}$ ,  $w_{edge}$  and  $w_{term}$  are appropriate weighting coefficients.

The line functional is defined by the image intensity itself and takes only the grey values into account, so

$$E_{line}(v(s)) = -I(x_s, y_s) \quad (2.4)$$

If  $w_{line}$  is large and positive, the snake will be attracted to the light lines, i.e. to locations

where  $I(x_s, y_s)$  is large.

If  $w_{line}$  is large and negative then the snake will be attracted to the dark lines, i.e. to locations where  $I(x_s, y_s)$  is low.

The edge functional is a feature that forces the active contour to converge to edges. Therefore, an edge should lead to a low energy potential. The edge functional is defined as

$$E_{edge}(v(s)) = -|\nabla I(x_s, y_s)|^2 \quad (2.5)$$

It attracts the snake to contours with large image gradients.

The termination functional detects sharp corners and ends of edges and lines by using the curvature of level lines in the image. If  $x = x(t)$  and  $y = y(t)$  then the curvature  $k$  is defined by

$$k \equiv \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} = \frac{\frac{d\phi}{dt}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}} = \frac{\phi'}{\sqrt{x'^2 + y'^2}} \quad (2.6)$$

where  $\phi$  is the tangential angle and  $s$  is the arc length, along the edge or line, and primed quantities are derivatives with respect to the parameter of the curve. Since

$$\tan \phi = \frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{y'}{x'} \quad (2.7)$$

we may write

$$\frac{d}{dt}(\tan \phi) = \frac{x'y'' - y'x''}{x'^2} \quad (2.8)$$

We also have

$$\frac{d}{dt}(\tan \phi) = (1 + \tan^2 \phi) \frac{d\phi}{dt} \quad (2.9)$$

Therefore,

$$\frac{d\phi}{dt} = \frac{1}{1 + \tan^2 \phi} \frac{x'y'' - y'x''}{x'^2} = \frac{x'x'' - y'y''}{x'^2 + y'^2} \quad (2.10)$$

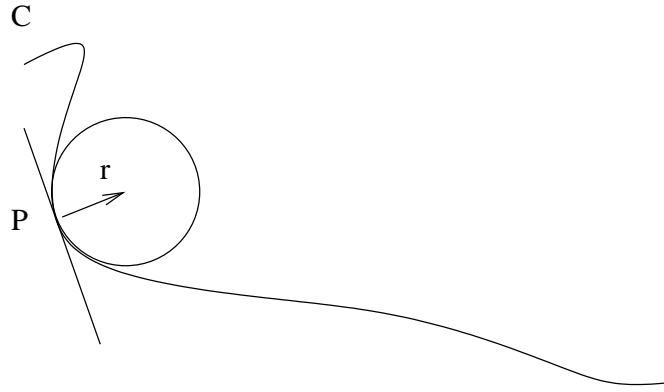


Figure 2.4: An osculating circle of a curve

So, from equations (2.6) and (2.10)

$$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}} \quad (2.11)$$

If  $y = f(x)$ , the equation of the curvature becomes

$$k = \frac{y''}{(1 + y'^2)^{3/2}} \quad (2.12)$$

since here  $s \equiv x$  and so  $x' = 1$  and  $x'' = 0$ .

For a plane curve  $C$ , the curvature at a point  $p$  has a magnitude equal to  $\frac{1}{r}$ , where  $r$  is the radius of an osculating circle (a circle that closely touches the curve at the given point and which has the same tangent as the curve) (see figure 2.4). The smaller the radius  $r$  of the circle, the larger  $\frac{1}{r}$  will be. So at places where a curve is almost straight, the curvature is near zero.

For a two-dimensional curve given implicitly by  $C(x, y) = 0$ , we can write

$$\frac{d}{dx}(C(x, y)) = \frac{\partial C}{\partial x} \frac{dx}{dx} + \frac{\partial C}{\partial y} \frac{dy}{dx} = 0 \Rightarrow C_x + C_y y' = 0 \quad (2.13)$$

By differentiating both sides of this equation with respect to  $x$  once more and remembering

that  $\frac{dC_x}{dx} = \frac{\partial C_x}{\partial x} + \frac{\partial C_x}{\partial y} \frac{dy}{dx}$  and  $\frac{dC_y}{dx} = \frac{\partial C_y}{\partial x} + \frac{\partial C_y}{\partial y} \frac{dy}{dx}$ , we obtain

$$C_{xx} + C_{xy}y' + C_{yx}y' + C_{yy}y'^2 + C_y y'' = 0 \quad (2.14)$$

Solving for  $y''$  yields

$$y'' = -\frac{C_{xx} + 2C_{xy}y' + C_{yy}y'^2}{C_y} \quad (2.15)$$

As we know from equation (2.13)  $y' = -\frac{C_x}{C_y}$ , therefore

$$y'' = -\frac{C_{xx}C_y^2 - 2C_{xy}C_xC_y + C_{yy}C_x^2}{C_y^3} \quad (2.16)$$

So, from equations (2.12) and (2.16), the curvature  $k$  is

$$k = -\frac{C_{xx}C_y^2 - 2C_{xy}C_xC_y + C_{yy}C_x^2}{(C_x^2 + C_y^2)^{3/2}} \quad (2.17)$$

To encourage snakes to latch at the ends of edges and lines and at corners, where the curvature is high, we set  $E_{term}(v(s)) = -|k|$ :

$$E_{term(i)} = -\left| \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}} \right| \quad (2.18)$$

and

$$E_{term} \equiv \sum_{i=1}^n E_{term(i)} \quad (2.19)$$

We can see that the calculation of curvature involves the calculation of second order derivatives. Differentiation tends to enhance noise, so usually, before the image derivatives are computed, the image is smoothed by convolving it with a Gaussian filter. However, we are not going to do this for our images, because we are interested in small details. The whole purpose of the preprocessing stage was to enhance such details. Smoothing will undo all we achieved by preprocessing.

### External forces

The external force is measured by external constraints such as constraints imposed by the user manually or automatically using higher level shape knowledge.

#### 2.4.2 Practical application

The integral of a function can be written as

$$\int_a^b f(s) ds \approx \sum_{i=1}^n f(s_i) \quad (2.20)$$

where  $a \leq s_1, \dots, s_n \leq b$

Also the squares of the derivatives of function  $v$  can be written as

$$\left| \frac{\partial v_i}{\partial s} \right|^2 \approx |v_i - v_{i-1}|^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \quad (2.21)$$

$$\left| \frac{\partial^2 v_i}{\partial s^2} \right|^2 \approx |v_{i-1} - 2v_i + v_{i+1}|^2 = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \quad (2.22)$$

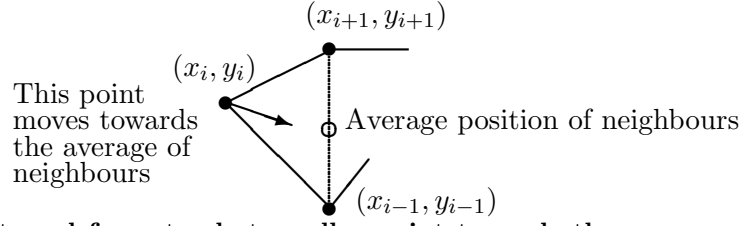
From equations (2.2), (2.20), (2.21) and (2.22) we obtain

$$\begin{aligned} E_{internal} = & \frac{1}{2} \sum_{i=1}^N \alpha_i ((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) + \\ & \frac{1}{2} \sum_{i=1}^N \beta_i ((x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2) \end{aligned} \quad (2.23)$$

where  $N$  is the number of control points and  $i$  is the index of a control point.

To minimise an energy, we need its gradient vector which constitutes the force that pulls the nodes of the snake at various positions:

$$F = \nabla E = (\nabla_x E, \nabla_y E) = \left( \frac{\partial E}{\partial x}, \frac{\partial E}{\partial y} \right) \quad (2.24)$$



**Figure 2.5:** The internal force tends to pull a point towards the average position of its neighbours

So the forces that correspond to the internal energy are

$$F_{x(internal(i))} = \alpha_i(x_i - x_{i-1}) - \alpha_{i+1}(x_{i+1} - x_i) + \beta_{i+1}(x_{i+2} - 2x_{i+1} + x_i) + 2\beta_i(-x_{i-1} + 2x_i - x_{i+1}) + \beta_{i-1}(x_{i-2} - 2x_{i-1} + x_i) \quad (2.25)$$

$$F_{y(internal(i))} = \alpha_i(y_i - y_{i-1}) - \alpha_{i+1}(y_{i+1} - y_i) + \beta_{i+1}(y_{i+2} - 2y_{i+1} + y_i) + 2\beta_i(-y_{i-1} + 2y_i - y_{i+1}) + \beta_{i-1}(y_{i-2} - 2y_{i-1} + y_i) \quad (2.26)$$

Equilibrium is reached when we are at a minimum of the energy, where the force becomes zero. It is clear then from equations (2.25) and (2.26) that this force becomes zero when  $x_i = x_{i+1}$  and  $x_i = x_{i-1}$ , as well as  $y_i = y_{i+1}$  and  $y_i = y_{i-1}$ . So this force tends to pull control point  $i$  towards both control point  $i - 1$  and control point  $i + 1$ , i.e. towards the middle of the last two points and so create a smooth snake (see figure 2.5).

According to equation (2.3), the image energy consists of three energies. The first one is:

$$E_{line} \equiv \sum_{i=1}^N E_{line(i)} = - \sum_{i=1}^N I(x_i, y_i) \quad (2.27)$$

The second term of image energy is represented by the gradient of the image:

$$E_{edge}(x, y) = -|\nabla I(x, y)|^2 = - \left( \left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2 \right) \quad (2.28)$$

As mentioned earlier, before computing the image derivatives, the image may first be smoothed with a Gaussian filter. In that case

$$C(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k g(i, j) I(x + i, y + j) \quad (2.29)$$

where,

$$g(i, j) = \frac{1}{Z} e^{-\frac{i^2 + j^2}{2\sigma^2}} \quad (2.30)$$

is a Gaussian filter of standard derivation  $\sigma$  and size  $(2k + 1) \times (2k + 1)$ , with  $Z$  being an appropriately chosen normalising constant, to ensure that the sum of the values of the digital filter is 1.

$$E_{edge} \equiv \sum_{i=1}^N E_{edge(i)} = -\frac{1}{4} \sum_{i=1}^N |C(x_i + 1, y_i) - C(x_i - 1, y_i)|^2 + |C(x_i, y_i + 1) - C(x_i, y_i - 1)|^2 \quad (2.31)$$

$E_{term}$  is given by equation (2.18) with

$$\begin{aligned} C_{xx}(i) &= C(x_i + 1, y_i) - 2C(x_i, y_i) + C(x_i - 1, y_i) \\ C_{yy}(i) &= C(x_i, y_i + 1) - 2C(x_i, y_i) + C(x_i, y_i - 1) \\ C_{xy}(i) &= C(x_i + 1, y_i + 1) - C(x_i + 1, y_i - 1) - C(x_i - 1, y_i + 1) + C(x_i - 1, y_i - 1) \\ C_x(i) &= \frac{1}{2}(C(x_i + 1, y_i) - C(x_i - 1, y_i)) \\ C_y(i) &= \frac{1}{2}(C(x_i, y_i + 1) - C(x_i, y_i - 1)) \end{aligned} \quad (2.32)$$

If no smoothing takes place, as is our case,  $C$  in these formulae is replaced by  $I$ .

The forces that correspond to the image energy term are given by:

$$F_{x(image(i))} = \nabla_x E_{image(i)} = w_{line} \nabla_x E_{line(i)} + w_{edge} \nabla_x E_{edge(i)} + w_{term} \nabla_x E_{term(i)} \quad (2.33)$$

$$F_{y(image(i))} = \nabla_y E_{image(i)} = w_{line} \nabla_y E_{line(i)} + w_{edge} \nabla_y E_{edge(i)} + w_{term} \nabla_y E_{term(i)} \quad (2.34)$$

where,

$$\nabla_x E_{line(i)} = -\frac{1}{2}(I(x_i + 1, y_i) - I(x_i - 1, y_i)) \quad (2.35)$$

$$\nabla_y E_{line(i)} = -\frac{1}{2}(I(x_i, y_i + 1) - I(x_i, y_i - 1)) \quad (2.36)$$

$$\begin{aligned} \nabla_x E_{edge(i)} = & \frac{1}{2}[E_{edge(x_i+1, y_i)} - E_{edge(x_i-1, y_i)}] = \\ & \frac{1}{2}[-\frac{1}{4}(|C(x_i + 2, y_i) - C(x_i, y_i)|^2 + \\ & |C(x_i + 1, y_i + 1) - C(x_i + 1, y_i - 1)|^2) + \\ & \frac{1}{4}(|C(x_i, y_i) - C(x_i - 2, y_i)|^2 + \\ & |C(x_i - 1, y_i + 1) - C(x_i - 1, y_i - 1)|^2)] \end{aligned} \quad (2.37)$$

$$\begin{aligned} \nabla_y E_{edge(i)} = & \frac{1}{2}[E_{edge(x_i, y_i+1)} - E_{edge(x_i, y_i-1)}] = \\ & \frac{1}{2}[-\frac{1}{4}(|C(x_i + 1, y_i + 1) - C(x_i - 1, y_i + 1)|^2 + \\ & |C(x_i, y_i + 2) - C(x_i, y_i)|^2) + \\ & \frac{1}{4}(|C(x_i + 1, y_i - 1) - C(x_i - 1, y_i - 1)|^2 + \\ & |C(x_i, y_i) - C(x_i, y_i - 2)|^2)] \end{aligned} \quad (2.38)$$

$$\nabla_x E_{term(i)} = \frac{1}{2}[E_{term(x_i+1, y_i)} - E_{term(x_i-1, y_i)}] \quad (2.39)$$

$$\nabla_y E_{term(i)} = \frac{1}{2}[E_{term(x_i, y_i+1)} - E_{term(x_i, y_i-1)}] \quad (2.40)$$

From equations (2.18) and (2.85),  $\nabla_x E_{term(i)}$  and  $\nabla_y E_{term(i)}$  can also be calculated.

In the absence of any external constraints, the total snake energy can be written as

$$E_{snake} = \sum_{i=1}^N (E_{internal}v(i) + E_{image}v(i)) \quad (2.41)$$

At the point of equilibrium, where  $E_{snake}$  is minimum, the force vector must be zero:

$$F_{snake(i)} = F_{internal}v(i) + F_{image}v(i) = 0 \quad (2.42)$$

The  $x$  and  $y$  components of this vector are:

$$\begin{aligned} F_{x(snake(i))} = & \alpha_i(x_i - x_{i-1}) - \alpha_{i+1}(x_{i+1} - x_i) + \\ & \beta_{i-1}(x_{i-2} - 2x_{i-1} + x_i) + 2\beta_i(-x_{i-1} + 2x_i - x_{i+1}) + \\ & \beta_{i+1}(x_{i+2} - 2x_{i+1} + x_i) + F_{x(image(i))} = 0 \end{aligned} \quad (2.43)$$

$$\begin{aligned} F_{y(snake(i))} = & \alpha_i(y_i - y_{i-1}) - \alpha_{i+1}(y_{i+1} - y_i) + \\ & \beta_{i-1}(y_{i-2} - 2y_{i-1} + y_i) + 2\beta_i(-y_{i-1} + 2y_i - y_{i+1}) + \\ & \beta_{i+1}(y_{i+2} - 2y_{i+1} + y_i) + F_{y(image(i))} = 0 \end{aligned} \quad (2.44)$$

Or, after re-arranging:

$$\begin{aligned} F_{x(snake(i))} = & (\beta_{i-1})x_{i-2} + (-\alpha_i - 2\beta_i - 2\beta_{i-1})x_{i-1} + \\ & (\alpha_{i+1} + \alpha_i + \beta_{i+1} + 4\beta_i + \beta_{i-1})x_i + \\ & (-\alpha_{i+1} - 2\beta_{i+1} - 2\beta_i)x_{i+1} + (\beta_{i+1})x_{i+2} + \\ & F_{x(image(i))} = 0 \end{aligned} \quad (2.45)$$

$$\begin{aligned}
F_{y(snake(i))} = & (\beta_{i-1})y_{i-2} + (-\alpha_i - 2\beta_i - 2\beta_{i-1})y_{i-1} + \\
& (\alpha_{i+1} + \alpha_i + \beta_{i+1} + 4\beta_i + \beta_{i-1})y_i + \\
& (-\alpha_{i+1} - 2\beta_{i+1} - 2\beta_i)y_{i+1} + (\beta_{i+1})y_{i+2} + \\
F_{y(image(i))} = & 0
\end{aligned} \tag{2.46}$$

Each of the above two equations represents a set of linear equations, one for each of the  $n$  control points of the snake. So, each one of these sets of equations may be written in matrix form as

$$AX + F_{x(image)}(X, Y) = 0 \tag{2.47}$$

$$AY + F_{y(image)}(X, Y) = 0 \tag{2.48}$$

$$A = \begin{bmatrix}
c_1 & d_1 & e_1 & 0 & \cdots & 0 & a_1 & b_1 \\
b_2 & c_2 & d_2 & e_2 & 0 & \cdots & 0 & a_2 \\
a_3 & b_3 & c_3 & d_3 & d_3 & e_3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_{n-1} & 0 & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\
d_n & e_n & 0 & \cdots & 0 & a_n & b_n & c_n
\end{bmatrix} \tag{2.49}$$

where,

$$a_i = \beta_{i-1},$$

$$b_i = -\alpha_i - 2\beta_i - 2\beta_{i-1},$$

$$c_i = \alpha_{i+1} + \alpha_i + \beta_{i-1} + 4\beta_i + \beta_{i+1},$$

$$d_i = -\alpha_{i+1} - 2\beta_{i+1} - 2\beta_i,$$

$$e_i = \beta_{i+1}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.50)$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (2.51)$$

$$F_{x(image)} = \begin{bmatrix} F_{x(image(1))} \\ F_{x(image(2))} \\ \vdots \\ F_{x(image(n))} \end{bmatrix} \quad (2.52)$$

$$F_{y(image)} = \begin{bmatrix} F_{y(image(1))} \\ F_{y(image(2))} \\ \vdots \\ F_{y(image(n))} \end{bmatrix} \quad (2.53)$$

As mentioned above, the external energy depends on the image pixel values, so to solve equations (2.47) and (2.48), we can use an iterative method by making  $x$  and  $y$  change with time. At time  $t$ , before the snake reaches steady state,

$$AX_t + F_{x(image)}(X_t, Y_t) = F_x(t) \quad (2.54)$$

$$AY_t + F_{y(image)}(X_t, Y_t) = F_y(t) \quad (2.55)$$

where  $F_x(t)$  and  $F_y(t)$  are equal to minus the time derivatives of  $x$  and  $y$ . Therefore,

$$F_x(t) = -\gamma(X_t - X_{t-1}) \quad (2.56)$$

$$F_y(t) = -\gamma(Y_t - Y_{t-1}) \quad (2.57)$$

where  $\gamma$  is some parameter.

The position of the snake at time  $t$  is found by the forces acting on it. Since it is very difficult to find the image energy at time  $t$  and take it into account implicitly, we use the image force at time  $t - 1$

$$F_{x(image)}(X_t, Y_t) = F_{x(image)}(X_{t-1}, Y_{t-1}) \quad (2.58)$$

$$F_{y(image)}(X_t, Y_t) = F_{y(image)}(X_{t-1}, Y_{t-1}) \quad (2.59)$$

So

$$AX_t + F_{x(image)}(X_{t-1}, Y_{t-1}) = -\gamma(X_t - X_{t-1}) \quad (2.60)$$

$$AY_t + F_{y(image)}(X_{t-1}, Y_{t-1}) = -\gamma(Y_t - Y_{t-1}) \quad (2.61)$$

Then, the snake position at time  $t$  is given by

$$X_t = (A + \gamma I)^{-1}(\gamma X_{t-1} - F_{x(image)}(X_{t-1}, Y_{t-1})) \quad (2.62)$$

$$Y_t = (A + \gamma I)^{-1}(\gamma Y_{t-1} - F_{y(image)}(X_{t-1}, Y_{t-1})) \quad (2.63)$$

### Summary of this algorithm

- Initialize  $\alpha$ ,  $\beta$ ,  $w_{line}$ ,  $w_{edge}$ ,  $w_{term}$  and  $\gamma$ .
- Ask the user which image to load and load it.
- The user should put the control points on the image.
- Put the position of control points in two one dimensional arrays (one for the  $x$  co-ordinates and one for the  $y$  co-ordinates of the snake points).

- Calculate matrix  $A$ .
- Calculate  $(A + \gamma I)^{-1}$ .
- Calculate the image force for all snake points.
- The new positions of the control points of the snake are found from  

$$X_t = (A + \gamma I)^{-1}(\gamma X_{t-1} - F_{x(image)}(X_{t-1}, Y_{t-1}))$$
 and  

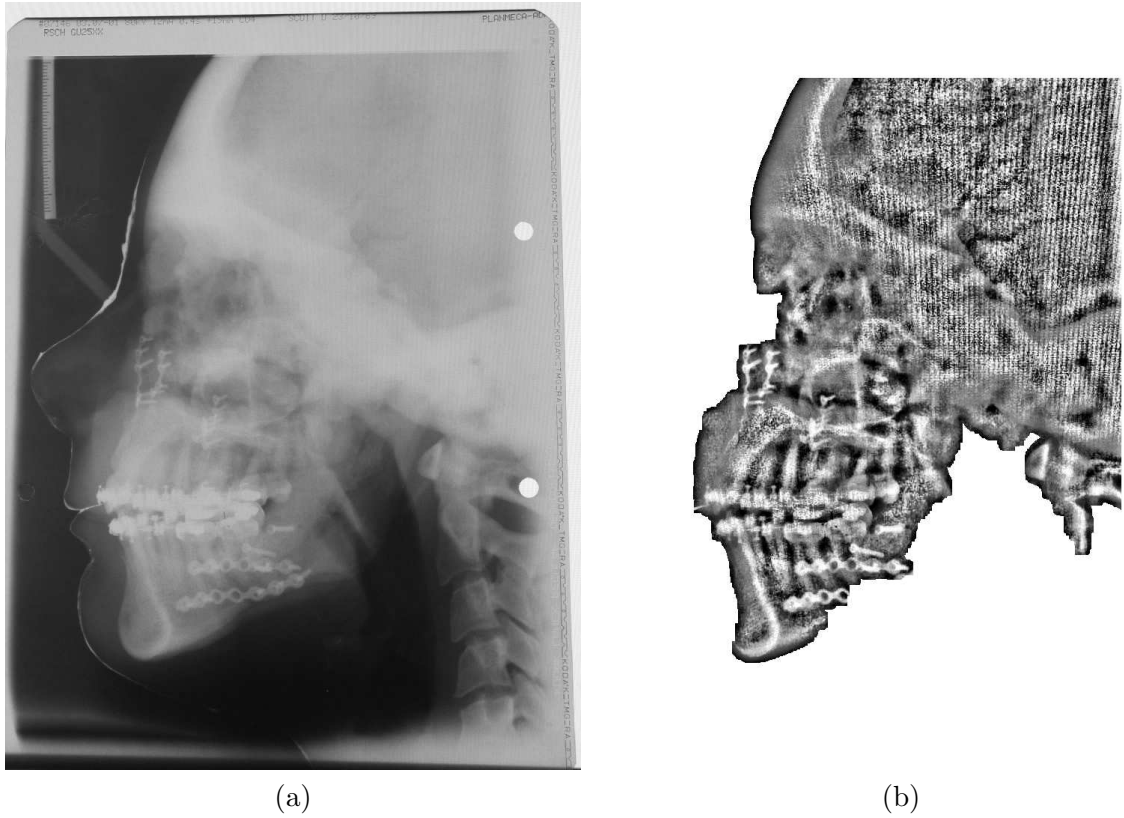
$$Y_t = (A + \gamma I)^{-1}(\gamma Y_{t-1} - F_{y(image)}(X_{t-1}, Y_{t-1}))$$
 where  $(X_{t-1}, Y_{t-1})$  are the previous positions.
- Show the new position of the snake on the image.
- Put these new positions in the arrays of the old positions.

### 2.4.3 Experiments

Here we used the X-ray image shown in figure 2.6(a). Experiments were carried out using the result of the pre-processing (see figure 2.6(b)) as an input image. By comparing the original X-ray image with the enhanced image (see figure 2.6(b)), it can be seen that it is more difficult to recognize the bone region (for example, nasion and orbitale) in the original X-ray image than in the enhanced one. Therefore, the pre-processing steps have been useful in influencing the working of the snake.

Figure 2.7(a) shows an initial contour placed in the enhanced image and figure 2.7(b) shows the position of the snake after minimising its energy. Figure 2.8 shows another snake and its final position. Finally, figures 2.9 and 2.10 show the application of this method to some CT images. Despite the pre-processing of the X-ray image, it is still more difficult to extract the bones from it than from CT images. Even so, when larger snakes were used, the result was disappointing. Such as example is shown in figures 2.8(a) and 2.8(b).

The parametric snake model can only handle topologically simple objects. This means that for this model to work, the topology of the object must be known in advance



**Figure 2.6:** (a) An X-ray image. (b) its enhanced version.

because the model can not deal with topological changes without implementing extra special procedures [49]. Therefore, the topology of the final curve will be the same as that of the initial curve. This will be a problem when an unknown number of objects must be detected at the same time, and contours have to be split or merged to fit them. Another problem of the parametric model is that the energy depends on the parameterisation of the curve and it does not depend on the object's geometry. That is why, Geometric Active Contours were proposed by Caselles et al. in 1993 [46].

## 2.5 Geometric active contours

This approach, also called “implicit active contours”, is based on curve evolution theory and the level set method. In this model, evolving curves use only a geometric measure which is independent of the parameterisation. It is best for recovery of objects with

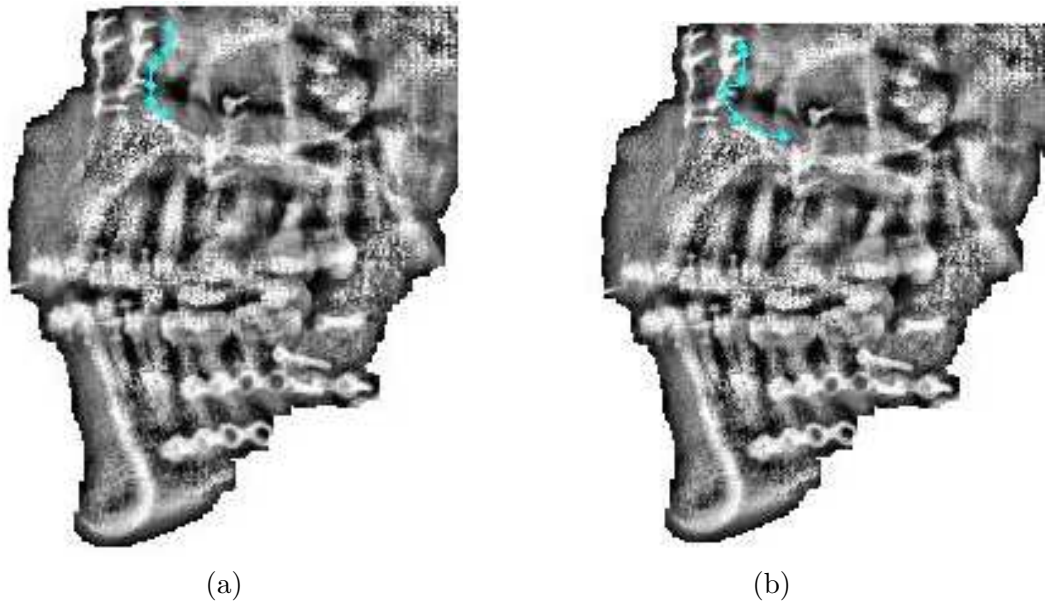


Figure 2.7: (a) Initial position of a snake. (b) Final position of the snake.

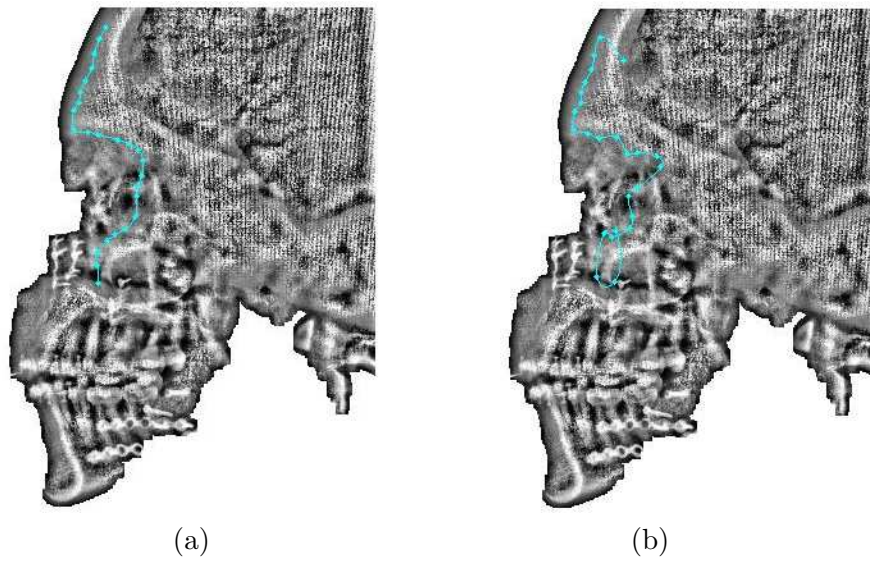


Figure 2.8: (a) Initial position of a snake. (b) Final position of the snake.

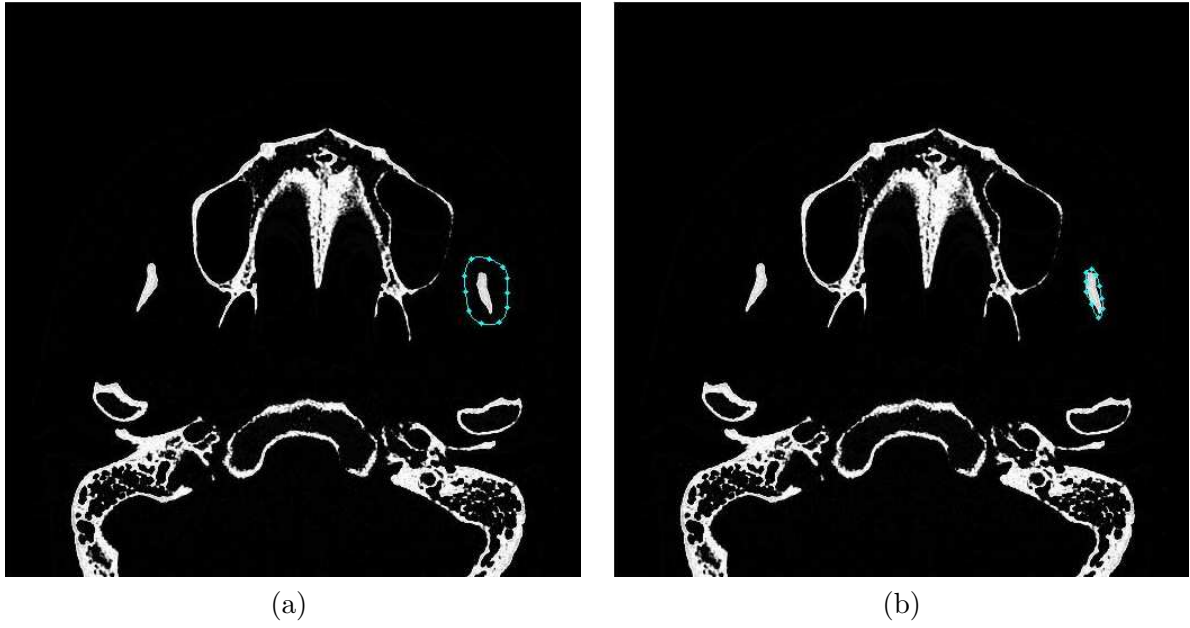


Figure 2.9: (a) Initial position of a snake. (b) Final position of the snake.

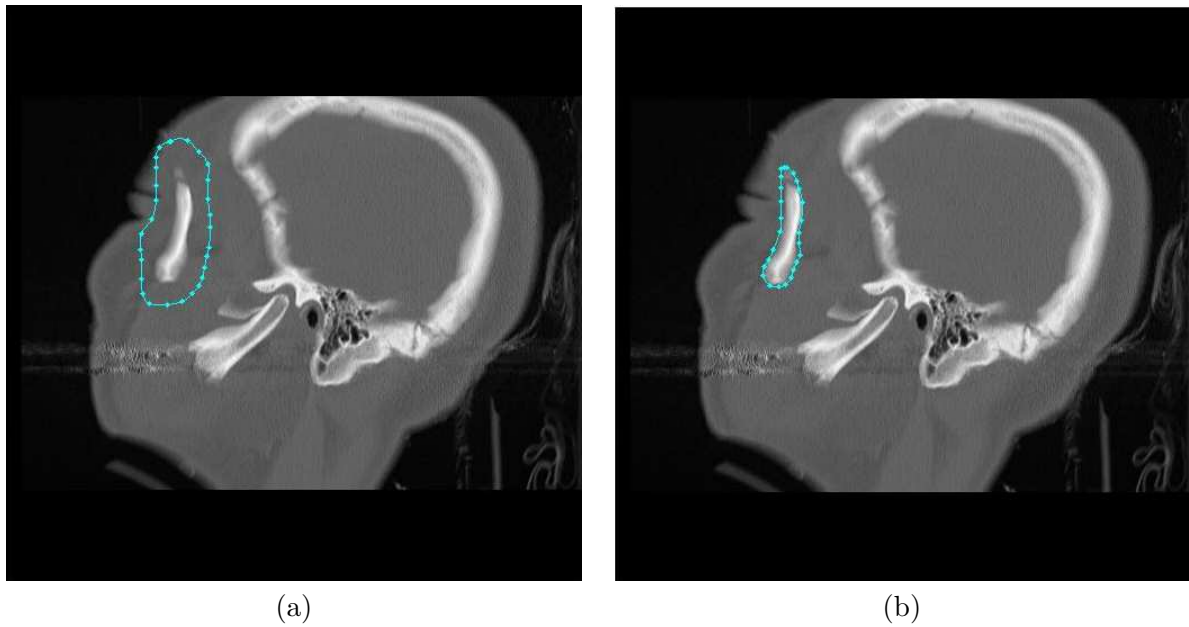


Figure 2.10: (a) Initial position of a snake. (b) Final position of the snake.

complex shapes and unknown topologies [47] [48].

In 2001, Chan and Vese [40] introduced a new geometric active contour based on the Mumford-Shah functional for segmentation [50], curve evolution and level set methods. Their model does not depend on the gradient of the image to stop the evolving curve and to detect object boundaries. Therefore, they called it “active contour without edges”. The main advantage of this model, which is used in this project, is that the position of the initial curve does not have to be near the object that is to be detected, and it can be anywhere in the image. To understand how this model works, first an introduction to level set methods is given.

### 2.5.1 Theory

#### Level Set Methods

Level set methods are numerical techniques which were first introduced by Osher and Sethian in 1988 [51]. Since then they have been used to track the evolution of interfaces (interfaces are curves in 2-D or surfaces in 3-D which separate the inside from the outside of some object) and shapes, in numerous applications such as robotics, image processing, computer vision, and computer graphics.

The central idea behind the standard level set method is to track the motion of an interface with a known speed function  $F$  as it evolves. So, let us assume an initial closed contour  $C_0(p) = (x(p), y(p))$  which is parametrized by  $p$  (say  $0 \leq p \leq 1$ ) in two dimensions. By evolving the contour in the direction of its Euclidean normal vector using a scalar speed function, a family of curves  $C(p, t) = (x(p, t), y(p, t))$  is generated, where  $t$  and  $p$  parametrize the family and each curve respectively (see figure 2.11). The evolution equation can be written as [52] [53]

$$\begin{cases} C_t(p, t) &= F(k)\bar{n} \\ C(p, 0) &= C_0(p) \end{cases} \quad (2.64)$$

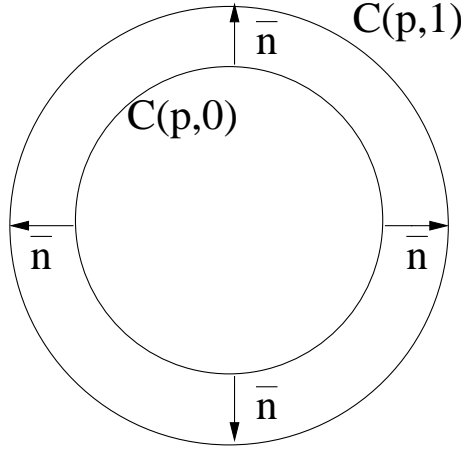


Figure 2.11: Curve evolution example

where

$$C_t(p, t) = \frac{\partial C(p, t)}{\partial t} \quad (2.65)$$

$F$  is the speed function of the curvature  $k$  of the curve and  $\bar{n}$  is the normal vector of the curve. This equation means that each point of the curve moves with the value of the speed function  $F$  at that point.

We may introduce a function  $\varphi(x, y, t)$  which expresses implicitly the dependence of curve  $C(p, t)$  on the image coordinates  $(x, y)$  and is defined so that it takes value 0 on curve  $C(p, t)$ :

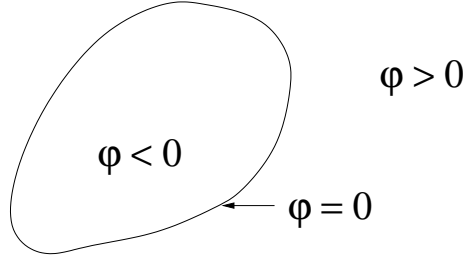
$$C(p, t) = \{(x, y) | \varphi(x, y, t) = 0\} \quad (2.66)$$

$$C(p, 0) = \{(x, y) | \varphi(x, y, 0) = 0\} \quad (2.67)$$

This function  $\varphi(x, y, t)$  is called level set function and it can be defined as the signed distance function from all  $(x, y)$  points to the curve points  $(\bar{X}_s)$ . Therefore, if  $\bar{X} = (x, y)^T \in \Re^2$ , then

$$\varphi(\bar{X}, t = 0) = \pm \min |\bar{X} - \bar{X}_s|, \bar{X}_s \in C(p, 0) \quad (2.68)$$

As it is shown in figure 2.12, the signed function  $\varphi$  takes negative values inside the curve and positive ones outside.

Figure 2.12: A signed distance function  $\varphi$ 

To find an equation of the motion function for the level set function, we consider that the level set value of a particle on the curve as a function of time must always be zero. So

$$\varphi(x, y, t) = 0, \quad \forall t \text{ if } (x, y) \in C(p, t) \quad (2.69)$$

By the chain rule,

$$\frac{\partial \varphi}{\partial t} + \frac{\partial \varphi}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \varphi}{\partial y} \frac{\partial y}{\partial t} = 0 \Rightarrow \frac{\partial \varphi}{\partial t} + \nabla \varphi \cdot \frac{d\bar{X}}{dt} = 0 \Rightarrow \varphi_t + \bar{\nu} \cdot \nabla \varphi = 0 \quad (2.70)$$

where  $\bar{\nu} \equiv \frac{d\bar{X}}{dt}$ .

For motion in the normal direction  $\bar{n} = \frac{\nabla \varphi}{|\nabla \varphi|}$ , we can use a scalar speed function  $F$ , so that  $\bar{\nu} = \frac{d\bar{X}}{dt} = F\bar{n} = F \frac{\nabla \varphi}{|\nabla \varphi|}$ . Then

$$\varphi_t + F |\nabla \varphi| = 0 \quad (2.71)$$

given the boundary condition that  $\varphi(X, t = 0)$  is known.

This equation describes the time evolution of the level set function where the propagating curve always identifies the zero level set of this function (see figure 2.13). In other words, it shows a connection between the family of moving curves  $C(p, t)$  and the family of evolving surfaces  $\varphi(x, y, t)$ . Therefore, at any time  $t$ , the curve is equal to the zero level set function  $\varphi$ . This means that the topology of curve  $C(p)$  may be changed as function  $\varphi$  evolves.

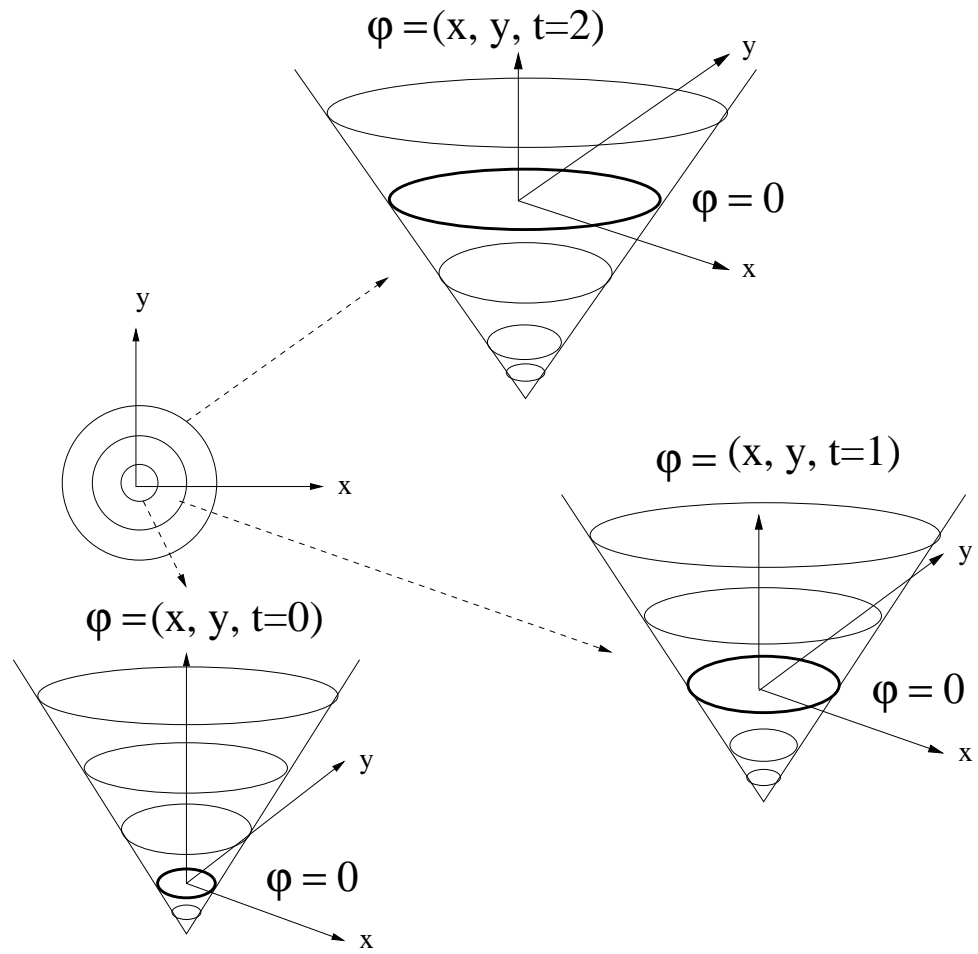


Figure 2.13: The curve of interest  $C$  at  $t = 0, 1$  and  $2$  is equal to the zero level set function  $\varphi$  at all these times [54]

### The Narrow Band Level Set Method

In this method,  $\varphi$  and  $F$  are updated at a small set of points in the neighborhood of the zero level set instead of being updated at all points. Only points inside a narrow strip around the curve are updated. When the zero level set reaches the edge of this strip, the strip is updated. The strip of the pixels, that are updated is defined as

$$\{X \text{ where } |\varphi(X)| < \delta\} \quad (2.72)$$

### Active contour without edges

As it was mentioned earlier, Chan and Vese [40] introduced an active contour model based on methods of curve evolution and level sets. Their method is the minimisation of an energy based segmentation  $E$ , which has a relation with the Mumford-Shah functional for segmentation [50].

$$\begin{aligned} E(C_1, C_2, C) = & \mu.length(C) + \nu.Area(inside(C)) + \\ & \lambda_1 \int_{inside(C)} |I - C_1|^2 dx dy + \\ & \lambda_2 \int_{outside(C)} |I - C_2|^2 dx dy \end{aligned} \quad (2.73)$$

where  $I$  is the input image,  $C_1$  and  $C_2$  are the mean intensities of the image  $I$  inside and outside the curve  $C$ , respectively, and  $\mu > 0, \lambda_1 > 0, \lambda_2 > 0, \nu = 0$  are constants<sup>1</sup>.

To find  $C$ ,  $C_1$  and  $C_2$  such that  $E(C_1, C_2, C)$  is minimised, the level set methods can be used in a way that curve  $C$  can be represented by the zero level set of signed function  $\varphi$ .

By using the Heaviside function  $H$ , and the Dirac measure  $\delta$ ,

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}, \quad \delta(z) = \frac{d}{dz} H(z) \quad (2.74)$$

we can write the energy function.

---

<sup>1</sup>In our implementation  $\mu = \lambda_1 = \lambda_2 = 1$

The gradient of the Heaviside function defines the curve, then the length of the curve ( $\varphi = 0$ ) can be computed as

$$length(\varphi = 0) = \int |\nabla H(\varphi)| dx dy = \int \delta(\varphi) |\nabla \varphi| dx dy \quad (2.75)$$

Also, the area of the inside curve is the integral of the Heaviside function of  $\varphi$ :

$$Area(\varphi = 0) = \int H(\varphi) dx dy \quad (2.76)$$

So,

$$\begin{aligned} E(C_1, C_2, C) = & \mu \int \delta(\varphi) |\nabla \varphi| dx dy + \nu \int H(\varphi) dx dy + \\ & \lambda_1 \int_{inside(C)} |I - C_1|^2 H(\varphi) dx dy + \\ & \lambda_2 \int_{outside(C)} |I - C_2|^2 (1 - H(\varphi)) dx dy \end{aligned} \quad (2.77)$$

also  $C_1$  and  $C_2$  are computed as:

$$C_1 = \frac{\int I H(\varphi) dx dy}{\int H(\varphi) dx dy} \quad (2.78)$$

$$C_2 = \frac{\int I (1 - H(\varphi)) dx dy}{\int (1 - H(\varphi)) dx dy} \quad (2.79)$$

The best constant values for  $C_1$  and  $C_2$  are the average intensities of the image pixels inside and outside of the curve, respectively.

Then, the evolution function derived from the minimisation of equation (2.77) is given by

$$\frac{\partial \varphi}{\partial t} = |\nabla \varphi| \left[ \mu \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} - \nu - \lambda_1 (I - C_1)^2 + \lambda_2 (I - C_2)^2 \right] \quad (2.80)$$

### 2.5.2 Practical Implementation

The discrete version of the level set equation (equation (2.71)) is

$$\frac{\varphi_{i,j}^{n+1} - \varphi_{i,j}^n}{\Delta t} + F |\nabla_{i,j} \varphi_{i,j}^n| = 0 \quad (2.81)$$

So,

$$\varphi_{i,j}^{n+1} = \varphi_{i,j}^n - F \nabla_{i,j} \varphi_{i,j} \Delta t \quad (2.82)$$

where  $\Delta t$  is the time step and

$$\nabla_{i,j} \varphi_{i,j} = \nabla \varphi(x_i, y_j) = \left( \frac{\partial \varphi_{i,j}}{\partial x}, \frac{\partial \varphi_{i,j}}{\partial y} \right) \quad (2.83)$$

In equation (2.80)  $\nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$  is the curvature of the curve

$$k = \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}} \quad (2.84)$$

where

$$\begin{aligned} \varphi_{xx} &= \frac{\partial^2 \varphi_{i,j}}{\partial x^2} = \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x \Delta x} \\ \varphi_{yy} &= \frac{\partial^2 \varphi_{i,j}}{\partial y^2} = \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta y \Delta y} \\ \varphi_{xy} &= \frac{\partial^2 \varphi_{i,j}}{\partial x \partial y} = \frac{\varphi_{i+1,j+1} - \varphi_{i+1,j-1} - \varphi_{i-1,j+1} + \varphi_{i-1,j-1}}{4\Delta x \Delta y} \\ \varphi_x &= \frac{\partial \varphi_{i,j}}{\partial x} = \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x} \\ \varphi_y &= \frac{\partial \varphi_{i,j}}{\partial y} = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y} \end{aligned} \quad (2.85)$$

The algorithm has four main steps:

- Create a curve (for example, a circle)
- Initialize  $\varphi$  so that its size is equal to the size of the image. This array can be initialized by finding the distance of all  $(x, y)$  points from the nearest point of the curve to them (see equation (2.68)). The value of  $\varphi$  is negative inside the curve and positive outside the curve.
- Compute  $C_1$  and  $C_2$

$$C_1 = \frac{\sum_{i,j \in \varphi \leq 0} I(i, j)}{n} \quad (2.86)$$



Figure 2.14: Input image

$n$  = number of pixels inside the curve

$$C2 = \frac{\sum_{i,j \in \varphi > 0} I(i,j)}{m} \quad (2.87)$$

$m$  = number of pixels outside the curve

- For every pixel  $(i,j)$ , compute  $\varphi_{i,j}^{n+1}$  using equation (2.82)

### 2.5.3 Experiments

Figure 2.14 shows the input image. Also, figure 2.15 shows the initial values of  $\varphi(x, y, t = 0)$  which are equal to the distance function of all  $(x, y)$  points from the curve. The size of  $\varphi$  is equal to the size of the image. Finally, figure 2.16 is the result of the algorithm.

Figure 2.17 shows the results of this algorithm on the original X-ray image and on the enhanced X-ray image. The result of the algorithm on two more CT images is shown in figure 2.18.

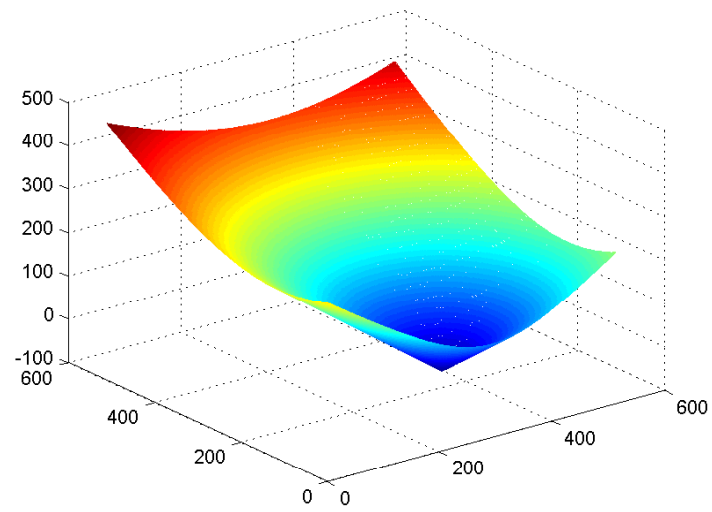


Figure 2.15:  $\varphi(x, y, t = 0)$

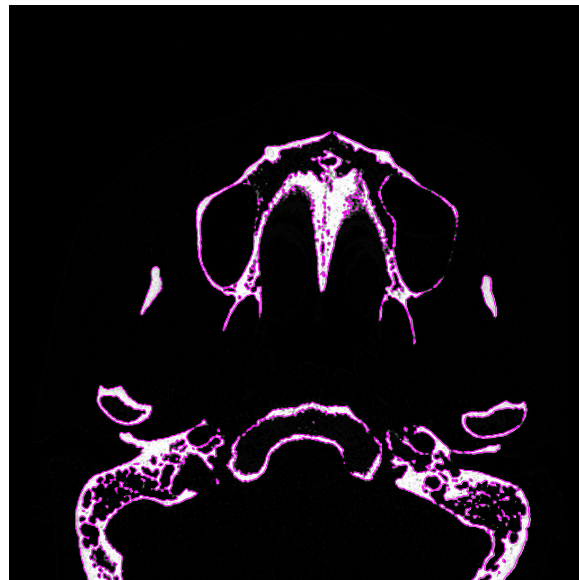
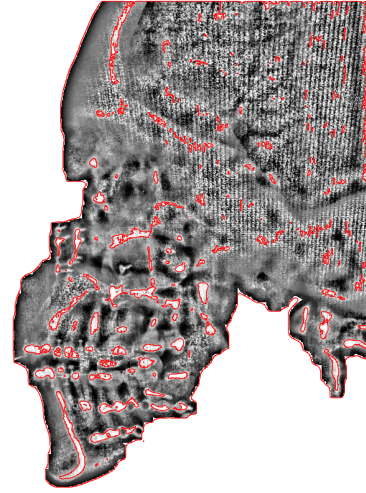


Figure 2.16: Final result (Output Image)

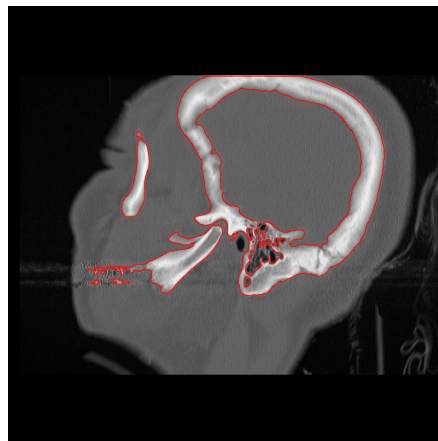


(a)

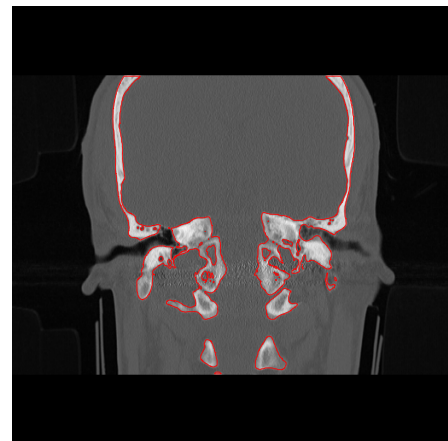


(b)

Figure 2.17: The results of the active contour model without edge: (a) on the original X-ray image. (b) on the enhanced X-ray image.



(a)



(b)

Figure 2.18: The active contour model without edges applied to two different CT images

### 2.5.4 Discussion and conclusions

As it is shown from the implementation results, the geometric active contours model can extract bones accurately in CT images, despite variations in size and shape. In addition, although the extraction of the bone area from X-ray images is more complex than the extraction from the CT image, experiments showed that this method can give reasonable results even for the X-ray image. The algorithm allows the curve or contour to change its topology as it evolves. Therefore, it can detect more than one bone area at the same time. However, the parametric snake model can only handle topologically simple objects. This means that for this model to work, the topology of the object must be known in advance. This will be a problem when an unknown number of objects must be detected at the same time, and contours have to be split or merged to fit them.

From all the above reasons, the geometric active contour method may be used to segment the bones in CT images. Also, since we are looking for some specific bones in X-ray images, the snake model is the ideal model to identify bones in X-ray images. However, to get more efficient results, the snake model has to be improved by adding an extra energy term, as described in the next chapter.

## 2.6 Conclusions from this chapter

The extraction of the bones area from medical images is an important step in this project. This chapter discussed active contour models which are useful in segmenting the information of interest. First, the Kass algorithm, which is a parametric active contour, was described in detail. Then, the Chan and Vese algorithm, that is a geometric active contour, was described and implemented. From the results of these two algorithms, we conclude that the Kass algorithm is the ideal choice to identify cephalometric landmark, but the ideal choice to automatically segment bones from the CT images is the Chan and Vese algorithm. The shapes of the extracted bones in CT images can then be used to compare them to find abnormalities and develop a diagnostic tool which is discussed in Chapter 4.

## Chapter 3

# Cephalometric Landmarks Identification Using Probabilistic Relaxation

### 3.1 Introduction

**C**EPHALOMETRY is a scientific measurement of human head which was first introduced by Broadbent in 1931 [34]. Since then it has been largely utilized to study the bony structure of the face for the diagnosis of abnormality, surgical planning and treatment, and studies on craniofacial growth. It is based on a set of agreed upon feature points (Cephalometric landmarks or bony landmarks) from standardized lateral skull radiographs, or cephalogram.

Broadbent (1931) developed a standardized roentgenographic<sup>1</sup> technique to permit estimation of growth or of treatment response by using a head special holder. Later, in 1948, Dawns introduced the first cephalometric analysis method based on a study of a group of selected individuals. His work has been the basis for most recent methods such as Ricketts's method [55] and Steiner's method [56]. These techniques, which are based on the location of the landmarks, were used to measure lines between significant landmarks.

---

<sup>1</sup>Photography with the use of x-rays

The number of bony landmarks which are defined is quite large. However, according to Rakosi [35], only a small number of these landmarks (20-30) are frequently used. In addition, despite all the efforts (which are discussed later) to automate locating of landmarks, these automatic methods are not currently used in clinical routine because of high failure rate. Locating landmarks on an X-ray image by an experienced person is a time consuming process [36] [37] [57].

## 3.2 Landmark Errors

According to Baumrind and Frantz [58], there are two common groups of error related to cephalometric measurements. The first one is called “projection”, which happens because of the geometry of the radiographic setup. For example, blurring of the image is a common problem that depends on the size of the X-ray anode (optical blurring), motion of subject or machine [59]. The second group is called “errors of identification”, which is due to the identification of uncertain specific landmarks on the X-ray image. There are several factors which affect the accuracy of landmark identifications. El-Mangoury et al. considered the horizontal, vertical and radial variability of 13 landmarks and they found that each landmark has its own characteristics [60]. Landmarks laying on a sharp curve are easier to be found than those laying on a flat curve. Moreover, identifying points that are located in areas of high contrast is more effortless than identifying those located in areas of low contrast [58] [59] [60].

## 3.3 Previous Work

Locating landmarks manually is a tedious and time consuming task and can be affected by human error. In addition, after physical radiographs have been replaced by on screen digital images, it is no longer possible to trace landmarks manually from X-ray images. Hence, cephalometric analysts utilize semi-automatic systems to plot landmarks directly to the screen. Consequently, there have been a number of attempts to automate landmark location and several researchers have studied the difficulty of this issue. Most of the

research can be categorized in two main categories: prior knowledge-based and learning [61] [62].

### 3.3.1 Prior Knowledge-Based methods

In 1986, Levy-Mandel et al. [63] were the first researchers who attempted to automate the location of cephalometric landmarks using knowledge-based techniques. Their work was divided in three steps: line enhancement, line extraction and landmarking. At the first step, they used a median filter to enhance the contrast of the image and remove the noise. Then they tried to extract the edges by using a Mero-Vassy operator [64]. In the next stage, they used a knowledge-based line tracing algorithm, which was based on a system that utilized prior knowledge about the lines. The knowledge was divided in three categories according to the geometric definition of the lines: the approximate line position, where the line starts and stops, and the number and characteristics of each segment that make up the line, such as its approximate length and its starting and stopping points. The test on two high-quality cephalograms as the system inputs showed that 23 out of 36 landmarks could be found. However, they were not comparable in accuracy with an expert result [63].

Parthasarathy et al. in 1989 [65] improved the Levy-Mandel system by including a four-level resolution pyramid<sup>2</sup> to increase the speed of the system. Their algorithm relied on anatomical knowledge of the human facial characteristics to find features that contain the cephalometric landmarks. They used five cephalograms with different quality and compared the accuracy of the algorithm with results gained from two experts. The results of obtaining nine landmarks showed that 18% were located to within 1 mm, 58% within 2 mm. According to Forsyth et al. [36], an error of at most  $\pm 1$  mm is desirable and according to Rakosi [35], an error of  $\pm 2$  is acceptable when locating landmarks.

Tong et al. in 1990 [66] followed the previous work by looking for 17 landmarks, some on bony features and others on soft tissue. They applied a resolution pyramid,

---

<sup>2</sup>The resolution pyramid is a form of image representation that consists of a series of images of decreasing spatial resolutions and sizes. The original image forms the basis of the pyramid and the higher levels of the pyramid are low-pass-filtered and under sampled versions of the lower levels [65].

edge-enhancement, thresholding, and edge detection techniques to locate these points. In 1994, Davis and et al. [67] presented a knowledge based system which was based on a blackboard architecture [68]. Yan et al. [69], Jackson et al. [70] and Cohen et al. [71] [72] also presented similar methods based on knowledge based techniques.

There are several disadvantages of knowledge based systems. They are highly dependent on the quality of the X-ray images and they can only identify a landmark if it is located on an edge. Besides, it is difficult for a user to add new landmarks to the system without formulating a new rule.

### 3.3.2 Learning

#### Pattern Matching

In 1994, Cardillo et al. [73] presented a pattern matching technique based on grey-scale mathematical morphology to identify the locations of the landmarks. Their algorithm was tested on 40 X-ray images. At first, the algorithm tried to locate the landmarks on new X-rays and then training was performed to teach the algorithm the shape and locations of the landmarks after processing each X-ray. The most promising results of the system that was trained to locate 20 landmarks, showed that 85% of the landmarks were located within 2 mm.

Grau et al. in 2001 [38] developed the previous work further by using a line detection model to search for significant lines and then apply a pattern detection algorithm based on mathematical morphology techniques. The system was tested on 20 images to detect 17 landmarks. The results showed some improvement over Cardillo and Sid-Ahmad methods [73] when tested on a smaller set of images.

Desvignes et al. [74] focused on the problem of finding an initial estimate of the location of the landmarks on an image by using an adaptive coordinate space where locations are registered. A training set was utilized to produce the initial estimate on a new image. Their results showed that the error of the estimated position comparing to the real position was about 4.5 mm. By considering the fact that the acceptable error is

$\pm 2$  mm, their errors were quite large.

### Neural Networks

In this category, the researchers used neural networks and fuzzy interface systems to locate the landmarks. In 1995 Uchino et al. [75] presented a fuzzy learning machine which might learn the relation between the mean gray level value of an input and the location of the landmark. They used several steps with which they zoomed into the landmark by considering a pyramidal representation of the input image. The error was an average 2.3 mm for a set of images that were not used for training. The problem with this method is that images should be registered in a common space, having the same size, rotation and shifts.

Later on, in 1999 Chen et al. [76] used neural networks and genetic algorithms to search for sub-images that contained the required landmarks. However, they did not report in their paper the accuracy of the landmark estimated position.

In 2002, Innes et al. [77] used a pulse coupled neural network (PCNN) to highlight key craniofacial features from digital X-rays. An averaging filter was applied to minimise noise prior to using the PCNN. They tested the method on three landmarks with increasing difficulty of identification and used a set of 109 X-ray images. The results showed the success rate of 36.7%, 88.1%, and 93.6% for the regions containing landmark sella, the nose, and the chin, respectively. They also mentioned that they did not find an automated procedure for determining good parameter values and their system which is capable of image smoothing and segmentation required a manual intervention to set the parameters.

Later, El-Feghi et al. [37] presented a method based on the use of a neuro-fuzzy system and template-matching technique. At first, they used a trained neuro-fuzzy system (NFS) to produce an initial estimate of the landmark location and then they tried to find the exact location of the landmark by applying a template-matching algorithm. The method was tested on 565 images to locate 20 landmarks. The results revealed improvement over previous work.

Although, these methods do not rely on strong edges, they are dependent on the

image appearance around each landmark.

### 3.4 Active Shape Models (ASMs)

Hutton et al. [78] evaluated the use of active shape models for automatic identification of cephalometric landmarks. Although, they mentioned that their method might be used to provide first estimated locations of the landmarks, they also concluded that their method was not sufficiently accurate for fully automated landmarking.

They selected 68 pre-treatment cephalograms randomly, regardless of quality, from patients of different ages, racial groups and malocclusions<sup>3</sup>. Those images were roughly  $750 \times 950$  pixels with 256 grey levels. Then, they rejected five of these images because of not having enough quality for identifying the landmarks either manually or automatically. They also used a drop-one-out scheme with the training set [79] to test the accuracy of the algorithm.

Their ASM method required a training set of hand-annotated images. They used the template shown in figure 3.1 to overlay each image in the training set after manual annotation by the expert. Sella, nasion, porion, orbitale, ANS, PNS, point A, point B, pogonion, gnathion, menton, gonion, upper and lower incisor tips and root apices were the landmarks used in their study (see Appendix A). By applying principal component analysis (PCA), the coordinates for the points that defined each template were analysed. The Procrustes algorithm [80] was used to ensure that the templates were all aligned with each other. The algorithm computed the mean shape and aligned all the examples to it until convergence. The ASM combined a global shape model with a local search at each point. For a new image, the mean template was overlaid onto the image and then a local search was performed at each point to find whether there was a nearby point in the image that matched to the intensity profile model learned from the training set (they used Cootes et al. [81] method). On average, 35 percent of the landmarks were within a range of  $\pm 2$  mm.

---

<sup>3</sup>Refers to an abnormal fit of the upper and lower dental structures

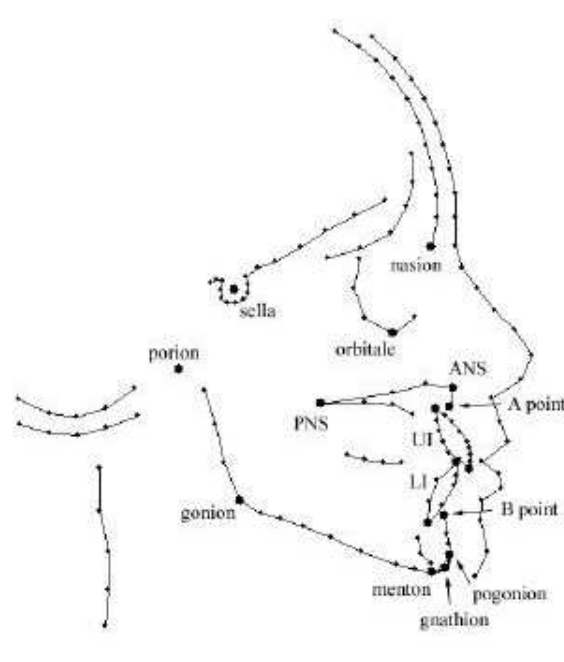
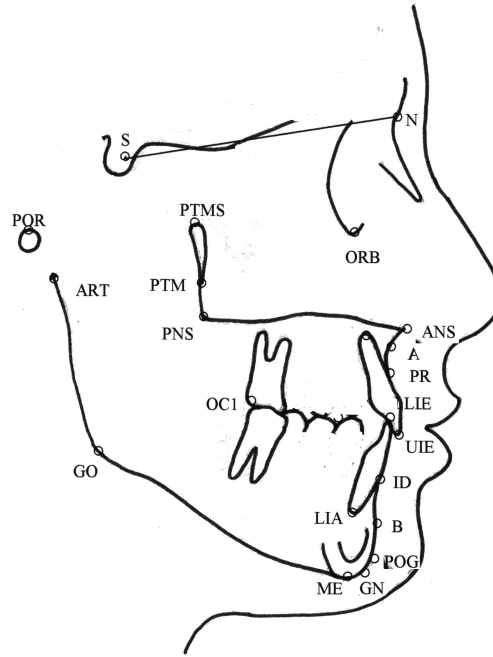


Figure 3.1: The cephalometric tracing used for the shape template (taken from [78]).

A lot of the methods used in the past failed because they tried to identify the landmarks in isolation. Some of the most promising methods are those that use the whole template to match, thus taking into consideration global contextual information. In this chapter, we develop methodology to locate cephalometric landmarks, supplied as a shape template shown in figure 1.1, repeated here for convenience as figure 3.2, on X-ray images, in an automatic way. In particular, this chapter focuses on developing a method based on probabilistic relaxation to identify the location of these landmarks on X-ray images. We consider the shape template as a separate image (the model) which contains some components of an X-ray image (the scene) and then try to find a correspondence between them.

### 3.5 Theoretical Background

Probabilistic relaxation labeling is an iterative estimation (a parallelisable process) for obtaining labels for a set of components by using contextual information. The algorithm



**Figure 3.2: Cephalometric landmarks**

was first introduced by Rosenfeld et al. in 1976 [82] for reducing or eliminating ambiguity of assigning labels to components of a scene. Later, in 1982, Rosenfeld and Kak used the method in pattern recognition and image segmentation [83]. Since then, many researchers have developed the method for different problems in computer vision, such as classification, shape matching and feature enhancement [84] [85] [86] [87] [88] [89]. Our method is based on the work done by Christmas et al. which was developed for structural matching in 1995 [90].

The theory of probabilistic relaxation labeling can be stated as follows. Consider an image as a scene and another one as a model so that the scene contains some components of the model (by components we mean edges, junctions etc.). By defining unary measurements, which are related to a single component, and binary relational measurements, that are related to pairs of components, and using prior knowledge extracted from the model, the probabilistic relaxation method can find a good matching between model and scene. So, let us consider the set of the scene components and the set of the model

components having  $N$  and  $M$  members, respectively. Each scene component is assigned a label  $\tau_i$  ( $i = 1, 2, \dots, N$ ), which takes its values from the model label set  $\Omega$ :

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_M\} \quad (3.1)$$

The notation  $\omega_{\tau_i}$  will be used to show a model label related with a particular scene label  $\tau_i$ .

Let us also define two sets of indices:

$$N_0 \equiv \{1, 2, \dots, N\} \quad (3.2)$$

$$N_i \equiv \{1, 2, \dots, i-1, i+1, \dots, N\} \quad (3.3)$$

For each component of the scene, there is a set of  $m_1$  measurement  $\mathbf{x}_i$  corresponding to the unary attributes of the component (such as length, orientation, or color):

$$\mathbf{x}_{i,i \in N_0} = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(m_1)}\} \quad (3.4)$$

The set of all unary measurement vectors  $\mathbf{x}_{i,i \in N_0}$  made on all components can be written as

$$\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad (3.5)$$

All the unary measurements  $m_1$  are also made on the model labels,

$$\check{\mathbf{x}} = \{\check{\mathbf{x}}_1, \check{\mathbf{x}}_2, \dots, \check{\mathbf{x}}_M\} \quad (3.6)$$

In addition, the relations of each pair of the scene components are expressed as a set of  $m_2$  binary measurements  $A_{ij}$ , where binary measurements may be the relative

position (or size, or orientation, etc) of one component with respect to another:

$$A_{ij} = \{A_{ij}^{(1)}, A_{ij}^{(2)}, \dots, A_{ij}^{(m_2)}\} \quad (3.7)$$

So, all the binary relations a scene component has with the other components in the set are:

$$A_{ij,j \in N_i} = \{A_{i1}, \dots, A_{ii-1}, A_{ii+1}, \dots, A_{iN}\} \quad (3.8)$$

Similarly, a set of  $m_2$  binary measurements  $\check{A}_{\alpha\beta}$  of each pair of model labels  $\omega_\alpha$  and  $\omega_\beta$  is denoted as

$$\check{A}_{\alpha\beta} = \{\check{A}_{\alpha\beta}^{(1)}, \check{A}_{\alpha\beta}^{(2)}, \dots, \check{A}_{\alpha\beta}^{(m_2)}\} \quad (3.9)$$

Moreover, to be able to compare the scene and the model measurements, the measurements must be expressed in the same domain. Hence, a set of  $m_0$  measurements that relate the scene and the model coordinate systems are also considered.

$$\Phi = \{\phi^1, \phi^2, \dots, \phi^{m_0}\} \quad (3.10)$$

After all the measurements are defined and computed, the next step is to find the best model label for each component of the scene. To do this, the labeling problem is formulated in the framework of Bayesian probability theory. The analysis that follows has been taken from [90]. The most suitable model label for a scene component is  $\omega_{\tau_i}$  if

$$P(\tau_i = \omega_{\tau_i} | \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i}) = \max_{\omega_\lambda \in \Omega} P(\tau_i = \omega_\lambda | \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i}) \quad (3.11)$$

By using Bayes's formula, this can be rewritten as

$$P(\tau_i = \omega_{\tau_i} | \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i}) = \frac{p(\tau_i = \omega_{\tau_i}, \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i})}{p(\mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i})} \quad (3.12)$$

The right-hand side can be expanded using the theorem of total probability:

$$\begin{aligned}
P(\tau_i = \omega_\alpha | \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i}) &= \frac{\sum_{\omega_{\tau_1} \in \Omega} \cdots \sum_{\omega_{\tau_{i-1}} \in \Omega} \sum_{\omega_{\tau_{i+1}} \in \Omega} \cdots \sum_{\omega_{\tau_N} \in \Omega}}{\sum_{\omega_{\tau_1} \in \Omega} \cdots \sum_{\omega_{\tau_N} \in \Omega}} \\
\frac{p(\tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i})}{p(\tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i})} & \quad (3.13)
\end{aligned}$$

The joint probability density that appears in the above expression may be factorised as follows:

$$\begin{aligned}
&p(\tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, \mathbf{x}_{j,j \in N_0}, A_{ij,j \in N_i}) = \\
&p(\mathbf{x}_{j,j \in N_0} | \tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, A_{ij,j \in N_i}) \times \\
&p(\tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, A_{ij,j \in N_i}) \quad (3.14)
\end{aligned}$$

Assume that the unary measurements and binary relations are independent of each other. Therefore,

$$\begin{aligned}
&p(\mathbf{x}_{j,j \in N_0} | \tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, A_{ij,j \in N_i}) = \\
&p(\mathbf{x}_{j,j \in N_0} | \tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}) \quad (3.15)
\end{aligned}$$

Notice that the unary measurement  $\mathbf{x}_j$  is independent of all labelings except the labeling  $\tau_j = \omega_{\tau_j}$ , hence, the above equation can be simplified to

$$p(\mathbf{x}_{j,j \in N_0} | \tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, A_{ij,j \in N_i}) = \prod_{j \in N_0} p(\mathbf{x} | \tau_j = \omega_{\tau_j}) \quad (3.16)$$

The second factor on the right-hand side of equation (3.14) may be rewritten as:

$$\begin{aligned}
&p(\tau_1 = \omega_{\tau_1}, \cdots, \tau_i = \omega_{\tau_i}, \cdots, \tau_N = \omega_{\tau_N}, A_{ij,j \in N_i}) = \\
&p(A_{i1} | A_{i2}, \cdots, A_{ii-1}, A_{ii+1}, \cdots, A_{iN}, \tau_1 = \omega_{\tau_1}, \cdots, \tau_N = \omega_{\tau_N}) \times \\
&p(A_{i2} | A_{i3}, \cdots, A_{ii-1}, A_{ii+1}, \cdots, A_{iN}, \tau_1 = \omega_{\tau_1}, \cdots, \tau_N = \omega_{\tau_N}) \times \cdots \times \\
&p(A_{iN} | \tau_1 = \omega_{\tau_1}, \cdots, \tau_N = \omega_{\tau_N}) \times P(\tau_1 = \omega_{\tau_1}, \cdots, \tau_N = \omega_{\tau_N}) \quad (3.17)
\end{aligned}$$

The above equation can be simplified by assuming that the binary relations are independent of each other and by considering the fact that from  $\tau_i = \omega_{\tau_i}$ , nothing is obtained about  $\tau_j = \omega_{\tau_j}$ . So,

$$p(\tau_1 = \omega_{\tau_1}, \dots, \tau_i = \omega_{\tau_i}, \dots, \tau_N = \omega_{\tau_N}, A_{ij, j \in N_i}) = \left\{ \prod_{j \in N_i} p(A_{ij} | \tau_j = \omega_{\tau_j}, \tau_i = \omega_{\tau_i}) \hat{P}(\tau_j = \omega_{\tau_j}) \right\} \times \hat{P}(\tau_i = \omega_{\tau_i}) \quad (3.18)$$

where the abbreviation  $\hat{P}$  denotes the prior probability of a label. The obtained result from (3.16) and (3.18) into (3.14) and subsequently into (3.13) is:

$$P(\tau_i = \omega_{\tau_i} | \mathbf{x}_{j, j \in N_0}, A_{ij, j \in N_i}) = \frac{P(\tau_i = \omega_{\tau_i} | \mathbf{x}_i) Q(\tau_i = \omega_{\tau_i})}{\sum_{\omega_{\lambda} \in \Omega} P(\tau_i = \omega_{\lambda} | \mathbf{x}_i) Q(\tau_i = \omega_{\lambda})} \quad (3.19)$$

where

$$Q(\tau_i = \omega_{\alpha}) = \sum_{\omega_{\tau_1} \in \Omega} \dots \sum_{\omega_{\tau_{i-1}} \in \Omega} \sum_{\omega_{\tau_{i+1}} \in \Omega} \dots \sum_{\omega_{\tau_N} \in \Omega} \left\{ \prod_{j \in N_i} P(\tau_j = \omega_{\tau_j} | \mathbf{x}_j) p(A_{ij} | \tau_i = \omega_{\alpha}, \tau_j = \omega_{\tau_j}) \right\} \quad (3.20)$$

This equation can be simplified again as follows:

$$Q(\tau_i = \omega_{\alpha}) = \prod_{j \in N_i} \sum_{\omega_{\beta} \in \Omega} P(\tau_j = \omega_{\beta} | \mathbf{x}_j) p(A_{ij} | \tau_i = \omega_{\alpha}, \tau_j = \omega_{\beta}) \quad (3.21)$$

Now, by combining (3.19) and (3.21) in an iterative scheme, the updated probabilities of a match at level  $n + 1$  are computed as

$$P^{(n+1)}(\tau_i = \omega_{\tau_i}) = \frac{P^n(\tau_i = \omega_{\tau_i}) Q^{(n)}(\tau_i = \omega_{\tau_i})}{\sum_{\omega_{\lambda} \in \Omega} P^{(n)}(\tau_i = \omega_{\lambda} | \mathbf{x}_i) Q^{(n)}(\tau_i = \omega_{\lambda})} \quad (3.22)$$

where

$$Q^{(n)}(\tau_i = \omega_{\alpha}) = \prod_{j \in N_i} \sum_{\omega_{\beta} \in \Omega} P^{(n)}(\tau_j = \omega_{\beta}) p(A_{ij} | \tau_i = \omega_{\alpha}, \tau_j = \omega_{\beta}) \quad (3.23)$$

The initialised probabilities can be computed by using the unary relations only:

$$P^{(0)}(\tau_i = \omega_{\alpha}) = \frac{p(\mathbf{x}_i | \tau_i = \omega_{\alpha}) \hat{P}(\tau_i = \omega_{\alpha})}{\sum_{\omega_{\lambda} \in \Omega} p(\mathbf{x}_i | \tau_i = \omega_{\lambda}) \hat{P}(\tau_i = \omega_{\lambda})} \quad (3.24)$$

## 3.6 The Proposed Method

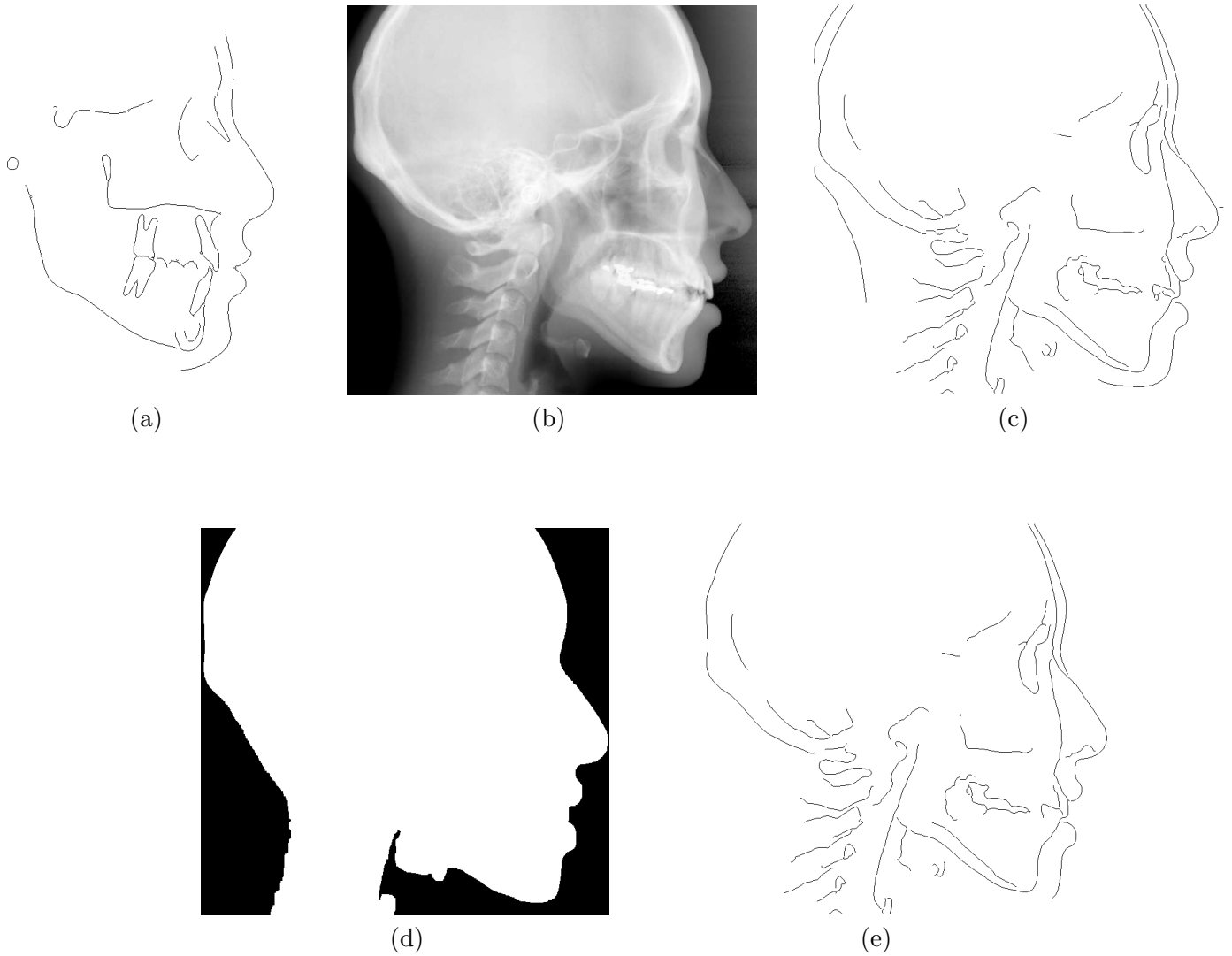
The theoretical background described in section 3.5 shows how the theory of probabilistic relaxation works in general. To develop this method for identifying the locations of cephalometric landmarks on X-ray images, we first consider the shape template as the model (see figure 3.3 (a)). Then, a preprocessing technique is required to extract a set of features from the facial X-ray image that can be comparable in type with the features from the model and also to reduce the amount of data in order to facilitate the process of matching. The image produced by the preprocessing method is considered as the scene.

### 3.6.1 Data Preprocessing

To locate the cephalometric landmarks on a facial X-ray image (see figure 3.3 (b)) by applying the probabilistic relaxation method, first we need to detect the most important features of the X-ray image. These are the edges. The edges are extracted by the Canny edge detection method. The extracted edges (see figure 3.3 (c)) include components that may be found in the model (see figure 3.3 (a)). To avoid noisy extracted edges in the background, we create a mask as described in the previous chapter. Only the area under the mask is kept for further processing and the rest of the image is disregarded. To create the mask, thresholding is used to convert the gray scale image to binary followed by mathematical morphology to remove small extra regions and holes (see figure 3.3 (d)). Then, the created mask is used on the edge map to remove all unwanted background edges. Also, the small edges (having a length smaller than 10 pixels) are removed (see figure 3.3 (e)).

### 3.6.2 A DFT-based technique for scaling measurement

In order to compare the scene and the model labeling, the measurements must be expressed in the same domain. To do this, the relative scale of the model and the scene, and the relative displacement should be found. So, after the data preprocessing stage, the next step is to scale the model and place it on the scene. To estimate the scale factor, a method



**Figure 3.3:** (a) Chephalometric landmarks image considered as the model. (b) An X-ray image. (c) Result after applying the Canny edge detection (mask size is  $17 \times 17$ ,  $\sigma = 2$ , and the lower and upper thresholds are 0.068 and 0.17, respectively). (d) Mask. (e) Result after using the mask on image (c). This result is considered as the scene.

based on discrete Fourier transform (DFT), log-polar mapping and phase correlation is used. Thus, first, a brief description of these is given below.

### A) Discrete Fourier Transform (DFT)

The DFT of an image  $I(x, y)$  of size  $M \times N$  is defined as follows [43]:

$$\hat{I}(m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^M \sum_{y=0}^N I(x, y) e^{-2\pi j(\frac{mx}{M} + \frac{ny}{N})} \quad (3.25)$$

Thus, the Fourier transform is a complex function that can be written in terms of its magnitude and phase:

$$\hat{I}(m, n) = \left| \hat{I}(m, n) \right| e^{j\phi_1(m, n)} \quad (3.26)$$

Also, the inverse discrete Fourier transform (IDFT) is written as

$$I(x, y) = \frac{1}{\sqrt{MN}} \sum_{m=0}^M \sum_{n=0}^N \hat{I}(m, n) e^{2\pi j(\frac{mx}{M} + \frac{ny}{N})} \quad (3.27)$$

### B) Log-Polar Mapping (LPM)

Let  $I(x, y)$  be an  $M \times N$  image matrix on the Cartesian plane  $(x, y)$ , and  $(x_c, y_c)$  the location of the central pixel.

$$\begin{cases} x_c = \frac{M}{2} & (if \ M \ odd) \\ x_c = \frac{M-1}{2} & (if \ M \ even) \end{cases} \quad \begin{cases} y_c = \frac{N}{2} & (if \ N \ odd) \\ y_c = \frac{N-1}{2} & (if \ N \ even) \end{cases} \quad (3.28)$$

Let  $r$  be the radial distance of a pixel location from the center  $(x_c, y_c)$  and  $\theta$  denote its orientation with respect to the  $x$ -axis (see figure 3.4), so

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (3.29)$$

$$\theta = \tan^{-1} \left( \frac{y - y_c}{x - x_c} \right) \quad (3.30)$$

Then, a pixel location  $p(r, \theta)$  in  $I$  can be defined as

$$p(r, \theta) \equiv (x(r, \theta), y(r, \theta)) \quad (3.31)$$

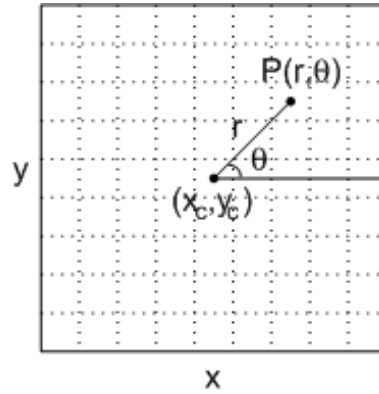


Figure 3.4: Discrete pixel location  $p(r, \theta)$

where

$$\begin{aligned} x(r, \theta) &= r \cos \theta + x_c \\ y(r, \theta) &= r \sin \theta + y_c \end{aligned} \quad (3.32)$$

Log-polar mapping is a technique which, instead of representing the pixel locations on the image in terms of  $(x, y)$  coordinates, represents the locations indexed by logarithmic distances  $\rho$  from the center of the image  $(x_c, y_c)$ , and angles  $\theta$ . In other words, the pixel location on the Cartesian plane  $(x, y)$  is converted to log-polar coordinates  $(\rho, \theta)$ :

$$\begin{aligned} \rho &= \log r \\ x &= 10^{\log r} \cos \theta + x_c \\ y &= 10^{\log r} \sin \theta + y_c \end{aligned} \quad (3.33)$$

However, computing  $(\rho, \theta)$  from the original rectangular grid leads to points that are not located exactly at points in the original grid. Thus, an interpolation method, such as nearest neighbour interpolation, is needed to find the values of the pixels in the log-polar plane  $(\tilde{I}(\rho, \theta))$ .

### C) Phase Correlation

Phase correlation is a frequency domain measurement method that expresses a shift in

the spatial domain as a phase shift in the frequency domain. Let  $I_1(x, y)$  and  $I_2(x, y)$  be two images which have a translating relation:

$$I_1(x, y) = I_2(x - x_0, y - y_0) \quad (3.34)$$

The relation of these two images in the frequency domain is equivalent to

$$\hat{I}_2(m, n) = \hat{I}_1(m, n)e^{-2\pi j(mx_0 + ny_0)} \quad (3.35)$$

The phase correlation is defined as the normalized cross power spectrum between  $\hat{I}_1(m, n)$  and  $\hat{I}_2(m, n)$ , which is [91]:

$$CPS(m, n) = \frac{\hat{I}_2(m, n)\hat{I}_1^*(m, n)}{|\hat{I}_2(m, n)\hat{I}_1^*(m, n)|} = \frac{|\hat{I}_1(m, n)| |\hat{I}_2(m, n)| e^{-j\phi_1} e^{j\phi_2}}{|\hat{I}_2(m, n)\hat{I}_1^*(m, n)|} = e^{-2\pi j(mx_0 + ny_0)} \quad (3.36)$$

where  $\hat{I}_1^*$  is the complex conjugate of  $\hat{I}_1$ .

The inverse Fourier transform of CPS is a sharp peak:

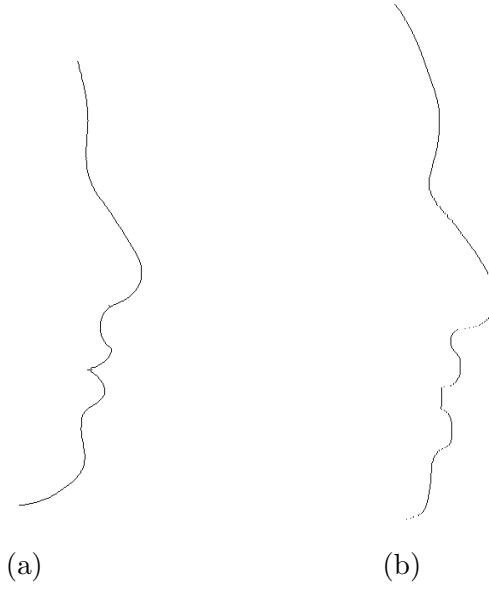
$$F^{-1}\{CPS(m, n)\} = \delta(x - x_0, y - y_0) \quad (3.37)$$

So, the location of the sharp peak gives the required shift  $(x_0, y_0)$ :

$$(x_0, y_0) = \arg\{\max_{(x, y)} \delta(x - x_0, y - y_0)\} \quad (3.38)$$

### 1. Scale factor estimation

To facilitate the scale measuring process between the scene and the model, the soft tissue surrounding the nose (from forehead to chin) is used as a reference feature. So, we first need to identify the soft tissue contour in both the scene and the model. We know the soft tissue landmark position in the model. However, its position in the scene must be identified and it is easier to find its approximate position from the mask rather than the scene. Therefore, to identify it, the mask image is searched from the right-hand side to find the first white point in each row. All these points show the location of the face



**Figure 3.5: (a) Soft tissue profile in the model. (b) Soft tissue profile in the scene.**

profile. After that, the soft tissue profiles in the model and the scene are considered as two separate images  $I_1$  and  $I_2$ , respectively (see figure 3.5). Let us assume that,

$$I_1(x, y) = I_2(sx, sy) \quad (3.39)$$

where  $s$  is the scale factor.

So, let  $\hat{I}_1(m, n)$  and  $\hat{I}_2(m, n)$  denote 2D DFTs of the images  $I_1(x, y)$  and  $I_2(x, y)$ , respectively. Then, equation (3.39) can be rewritten as [43]

$$\hat{I}_1(m, n) = \frac{1}{s^2} \hat{I}_2\left(\frac{m}{s}, \frac{n}{s}\right) \quad (3.40)$$

To find  $s$ , first the Cartesian coordinates  $(m, n)$  are converted to polar coordinates  $(r, \theta)$  where  $r$  denotes radial distance from the center  $(m_c, n_c)$  and  $\theta$  denotes angle:

$$r = \sqrt{(m - m_c)^2 + (n - n_c)^2} \quad (3.41)$$

$$\theta = \tan^{-1} \left( \frac{n - n_c}{m - m_c} \right) \quad (3.42)$$

$$r_2 = \sqrt{\left(\frac{m - m_c}{s}\right)^2 + \left(\frac{n - n_c}{s}\right)^2} = \frac{1}{s} \sqrt{(m - m_c)^2 + (n - n_c)^2} = \frac{r}{s} \quad (3.43)$$

$$\theta_2 = \tan^{-1} \left( \frac{\frac{n - n_c}{s}}{\frac{m - m_c}{s}} \right) = \tan^{-1} \left( \frac{n - n_c}{m - m_c} \right) = \theta \quad (3.44)$$

Thus, the magnitudes of  $\hat{I}_1(m, n)$  and  $\hat{I}_2(m, n)$  in the polar coordinate system can be written as

$$M_1(r, \theta) = \frac{1}{s^2} M_2 \left( \frac{r}{s}, \theta \right) \quad (3.45)$$

Then, the scaling can be reduced to a translation (the factor  $\frac{1}{s^2}$  is ignored)

$$M_1(\log r, \theta) = M_2(\log r - \log s, \theta) \quad (3.46)$$

This may be rewritten as

$$M_1(\rho, \theta) = M_2(\rho - b, \theta) \quad (3.47)$$

where  $\rho \equiv \log r$  and  $b \equiv \log s$

To obtain the scale factor  $s$ , the translation  $b$  is first estimated from the phase correlation technique (see equations (3.36) to (3.38)). So, the scaling  $s$  can be computed from  $b$  as:

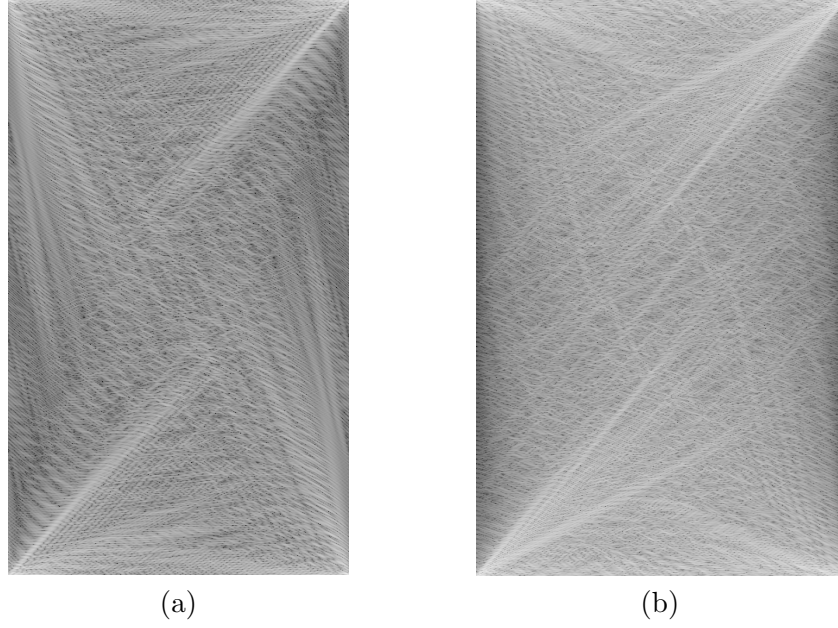
$$s = 10^b \quad (3.48)$$

## 2. In Practice

The sizes of the two images  $I_1$  and  $I_2$  are  $668 \times 395$  and  $743 \times 475$ , respectively. First, the 2D DFTs of  $I_1(x, y)$  and  $I_2(x, y)$  are computed to obtain  $\hat{I}_1(m, n)$  and  $\hat{I}_2(m, n)$ . The magnitude of the resultant DFTs are shown in figure 3.6. Then,  $\hat{I}_1(m, n)$  and  $\hat{I}_2(m, n)$  are zero padded to  $\tilde{I}_1(m, n)$  and  $\tilde{I}_2(m, n)$  with size  $K \times K$ :

$$K = \max(\text{size}(I_1), \text{size}(I_2)) = \max(668, 395, 743, 475) = 743 \quad (3.49)$$

Then, to represent the scaling factor as a shift, the magnitude values of padded DFTs are converted to log-polar coordinates. So, if a point  $(m, n)$  is related to the grid point



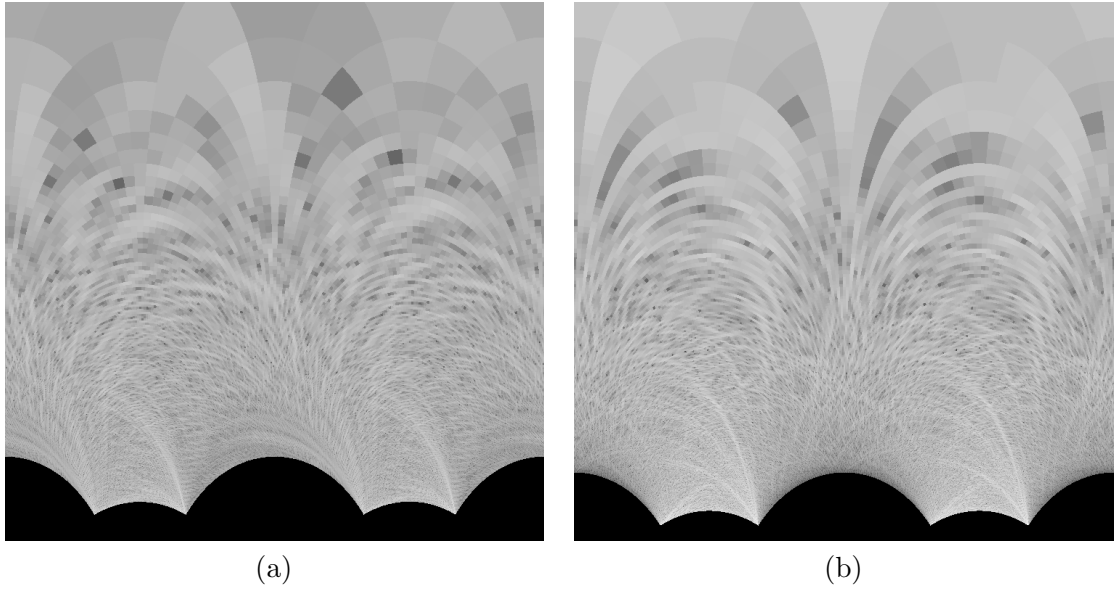
**Figure 3.6:** The magnitude of the Discrete Fourier of the images. (a) Model soft tissue landmark image. b) Scene soft tissue landmark image.

$(\log r, \theta)$ , we have

$$\begin{aligned} m &= 10^{\log r} \cos \theta + m_c \\ n &= 10^{\log r} \sin \theta + n_c \end{aligned} \quad (3.50)$$

By applying a polar coordinate transformation to  $|\tilde{I}_1(m, n)|$  and  $|\tilde{I}_2(m, n)|$ , radial lines in Cartesian space are mapped to vertical lines in the polar coordinate space. The results of the mapping are shown in figure 3.7 where  $\log r$  and  $\theta$  lie along the vertical and horizontal axes, respectively. Now, the image displacement ( $\log s$ ) can be estimated by using the inverse Fourier transform of the phase correlation technique (see equation (3.36) and (3.37)). The result is illustrated in figure 3.8. The location of the highest peak along the  $\rho$  axis is considered as  $\log s$  which in this example is equal to 15. The scale factor is estimated as following

$$r_{max} = \sqrt{\left(\frac{x_{max}}{2}\right)^2 + \left(\frac{y_{max}}{2}\right)^2} = \sqrt{\left(\frac{743}{2}\right)^2 + \left(\frac{743}{2}\right)^2} \simeq 525 \quad (3.51)$$



**Figure 3.7:** (a) The log-polar mapping of figure 3.6a. (b) The log-polar mapping of figure 3.6b.

$$\begin{cases} scale = 10^{\frac{\log(r_{max}-r_{min})}{x_{max}} \times position}; & \text{if } position \leq \frac{x_{max}}{2} \\ scale = \frac{1}{10^{\frac{\log(r_{max}-r_{min})}{x_{max}} \times (x_{max}-position)}}; & \text{if } position > \frac{x_{max}}{2} \end{cases} \quad (3.52)$$

where “position” here mean the position of the maximum peak in the  $(\rho, \theta)$  space, along the  $\rho$  axis.

In this example, the scale factor turns out to be 1.13. Figure 3.9 shows the scaled model which is placed on the X-ray input image and the scene. This may be also considered as the initial positioning of the cephalometric landmarks.

After the model components are scaled and placed on the input image, a global window is created around the current position of all key points (see figure 3.10). Then, the components outside the window are discarded. Moreover, to add more useful edges, local windows are created around the current position of each key point. Then the area inside each window is considered as a separated image and Canny edge detector method is applied to it (see figure 3.13 (b)).

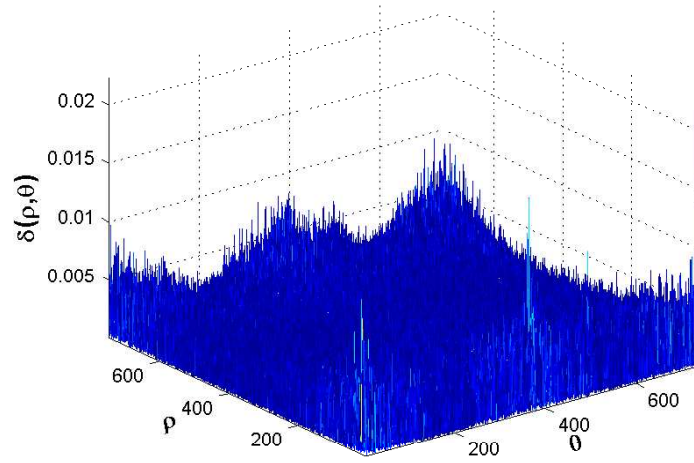


Figure 3.8: Phase correlation between the log-polar image shown in figure 3.7a and the log-polar image shown in figure 3.7b

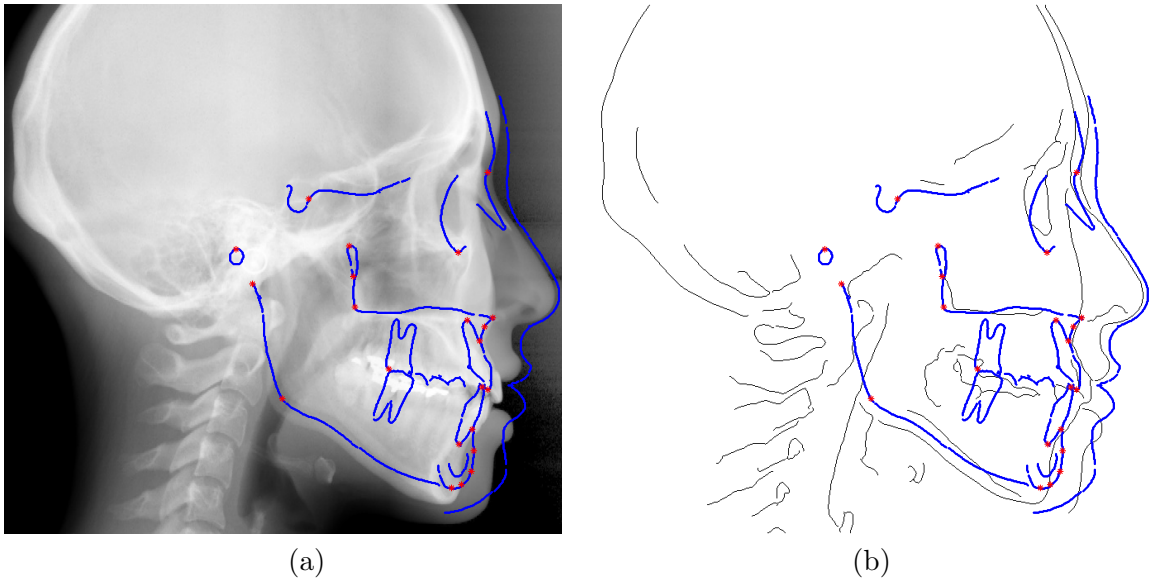
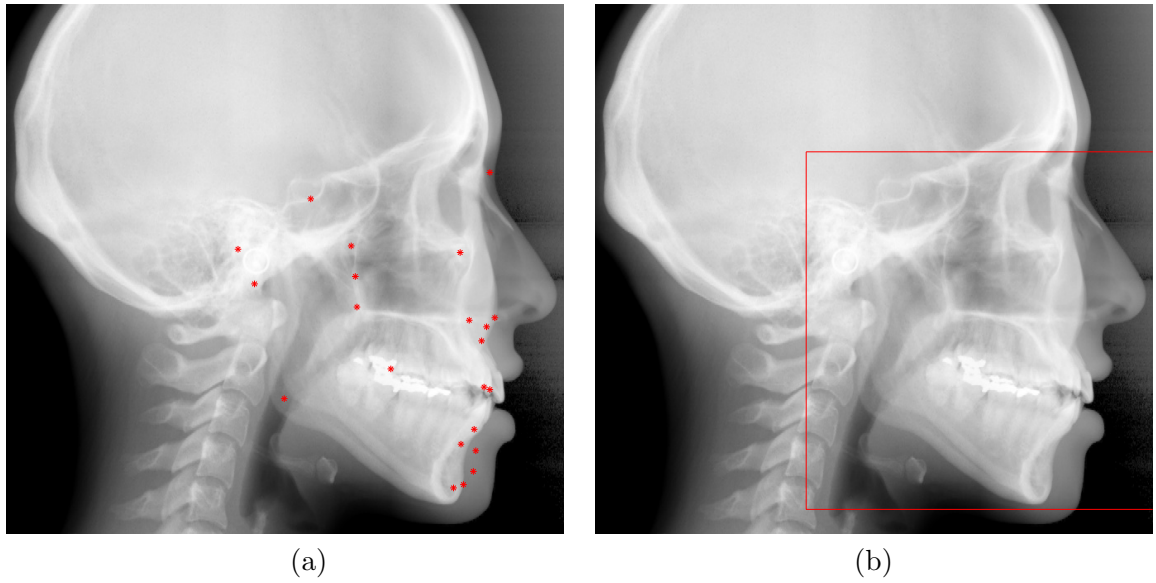


Figure 3.9: Initial positions of the cephalometric landmarks. (a) On the input image. (b) On its edge map.



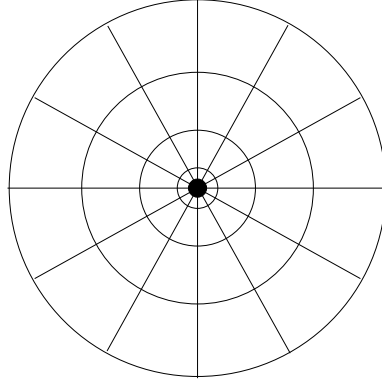
**Figure 3.10:** (a) Initial positions of the key cephalometric points on the input image. (b) Red rectangle indicates the created window surrounding the position of all key points.

### 3.6.3 Matching Procedure

The proposed matching method is based on the probabilistic relaxation method, and the shape context method [92] as a main measurement. The reason for utilizing the shape context method as the unary attribute of the component is that the shape context provides rich information about the local geometry of each component. It considers the relation between the component and the global shape (all components). Besides, the geometric quantity is invariant to scaling and translation. Thus, the advantage of the proposed method is that it considers all useful information of each component. So, first, a description of the shape context method is given.

#### What is a shape context?

Consider a set of points  $p_1, \dots, p_m$  which represent a component, such as a curve. The shape context of a point  $p_i$  is a histogram which expresses the relation of that point to the remaining points of the set. The histogram is built by dividing the space around that point into  $k$  bins and counting the number of points in each bin. The shape context of



**Figure 3.11: Diagram of log polar histogram bins**

point  $p_i$  is defined by

$$h_i(k) = \# \{p \neq p_i : (p - p_i) \in \text{bin}(k)\} \quad (3.53)$$

where  $\text{bin}(k)$  is uniform in a log-polar coordinate system. An example of this system with 4 bins for  $\log r$  and 12 bins for  $\theta$  is shown in figure 3.11.

The cost of matching two points  $p_i$  and  $q_j$  from two different sets is defined as

$$C_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (3.54)$$

where  $C_{ij}$  is the cost function, and  $h_i(k)$  and  $h_j(k)$  are the shape contexts of  $p_i$  and  $q_j$ , respectively.

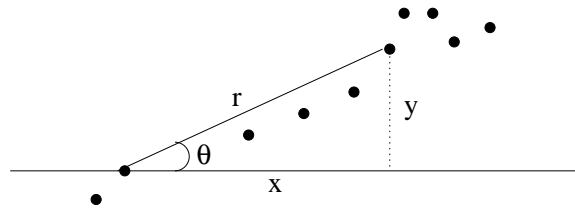
The following steps explain how to generate a shape context.

1) Consider the polar  $(r, \theta)$  coordinate system, where  $r$  and  $\theta$  are radial distance and angle, respectively. For each point, compute the distance  $r$  and angle  $\theta$  to all other points, with all angles being between 0 and  $2\pi$  (see figure 3.12):

$$r = \sqrt{x^2 + y^2} \quad \theta = \arctan \frac{y}{x} \quad (3.55)$$

2) After normalizing the distance by the mean distance, create bins that are uniform in log-polar space (for example figure 3.11 has 48 bins).

3) Quantize these distances and angles into the bins (see equation (3.53)). This means

Figure 3.12: Polar Coordinates  $r$  and  $\theta$ 

that for each point, put all other points in the appropriate bins according to their distances and angles.

### Measurements

As mentioned earlier, the model components are 22 cephalometric landmarks (see figure 3.13 (a)) and the scene components are the extracted edge points (see figure 3.13 (b)). Note that the cephalometric landmark S (see figure 3.2) is not located on any edge. Therefore, one of the points on its associated curve (the curve nearby) is considered as the key point (see figure 3.13 (a)).

In order to match these components (points), probabilistic relaxation is used. We start by characterising each landmark point with its shape context. We compute the shape contexts of the edge points in the scene and the shape contexts of the cephalometric landmarks (key points) in the model with respect to all curves. So, we define  $h_\alpha(k)$  to be the shape context of landmark  $\mathbf{x}_\alpha$  and  $h_i(k)$  to be the shape context of an edgel  $\mathbf{x}_i$ . The log-polar histogram is built with 60 bins (5 for  $\log r$  and 12 for  $\theta$ ).

An example of this measurement of a cephalometric landmark (key point) in the model is shown in figure 3.14. Figure 3.14 (a) shows one of the cephalometric landmarks on the model, and its shape context can be seen in figure 3.14 (b). The computed shape context is treated as the unary measurement of the point.

Note that after the model components are scaled, they are placed on the scene. This helps one to allow matches only within a window of  $N_{\tau_i} \times N_{\tau_i}$  around the current position of each component  $\tau_i$  (we consider the same size window for all components).

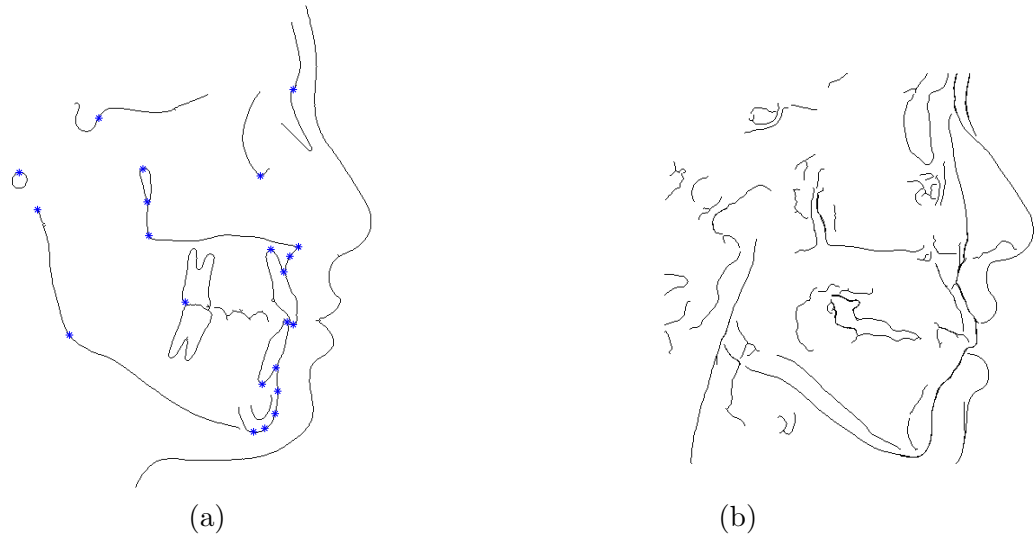


Figure 3.13: (a) Blue stars indicate cephalometric landmarks. (b) the scene components are the extracted edges.

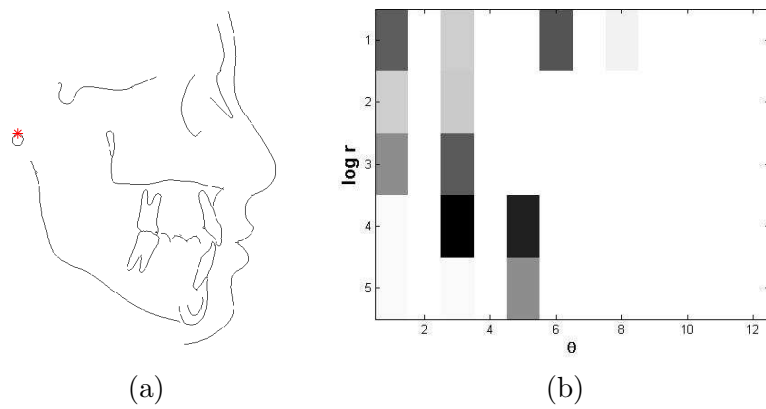


Figure 3.14: Unary measurements. (a) A red star indicates a landmark in the model. (b) Shape context of the red point with 60 bins (5 for  $\log r$  and 12 for  $\theta$ ) (black = large value).

Also, to prevent unnecessary computations, the extracted edge points which are not inside any windows are not considered in the further stages.

After computing the shape contexts for the components in the model and the scene, the next step is to find the cost function  $C(\mathbf{x}_i, \check{\mathbf{x}}_\alpha)$  of each pair of components in the model and the scene according to equation (3.54). The pairing of components with the lowest cost function has the highest probability.

Then,  $p(\mathbf{x}_i | \tau_i = \omega_\alpha)$  of the Christmas et al. [90] formalism replaced with

$$p(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha) = \frac{1}{C(\mathbf{x}_i, \check{\mathbf{x}}_\alpha)} \quad (3.56)$$

So, the probabilities are initialised as:

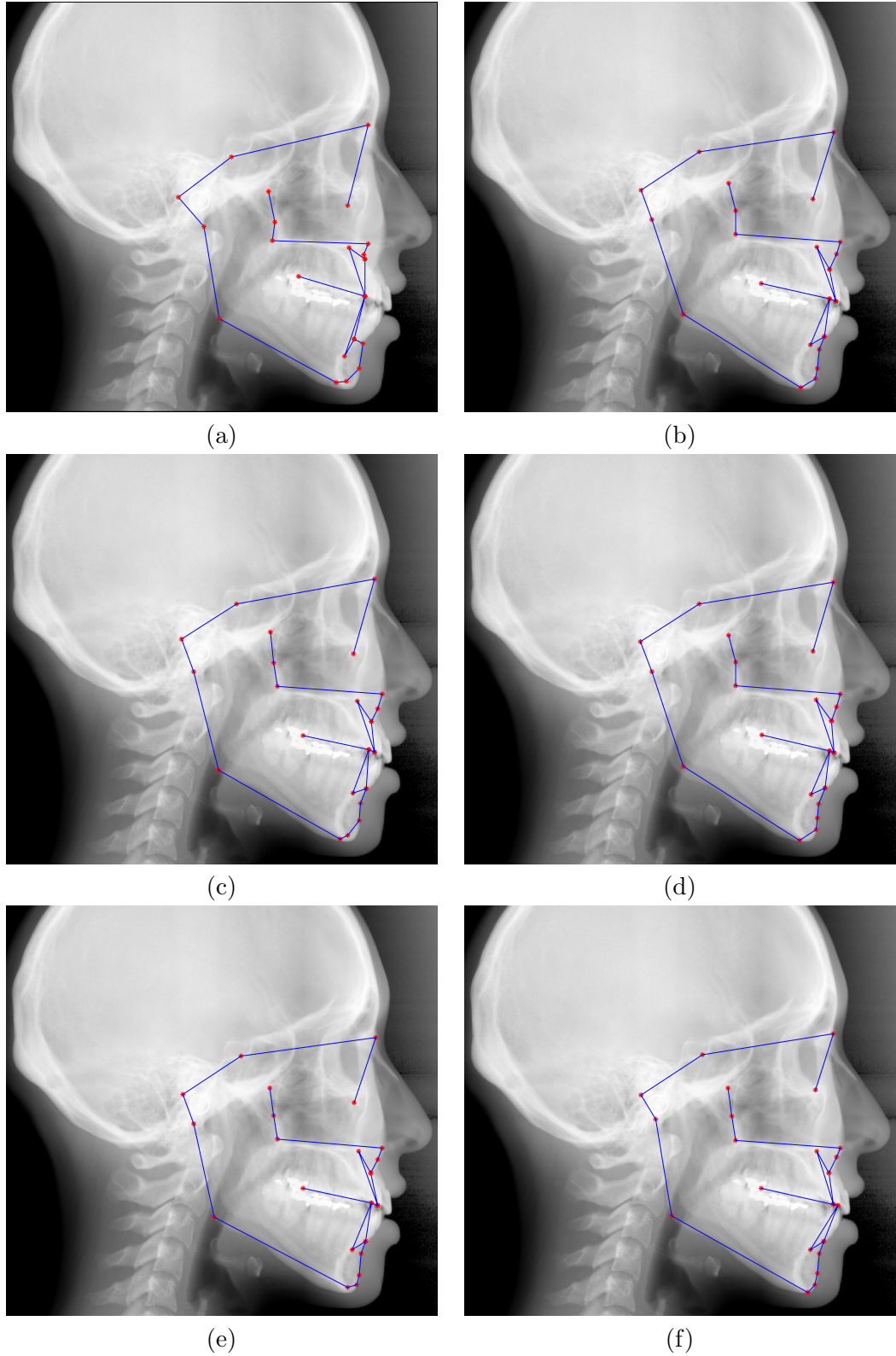
$$P^{(0)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha) \equiv \frac{p(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha)}{\sum_i p(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha)} \quad (3.57)$$

In addition, we use the relative position and orientation of the landmarks as contextual information to be incorporated by the method of probabilistic relaxation. In particular, the Euclidean distance  $r_{\alpha\beta}$  and relative orientation  $\theta_{\alpha\beta}$  between two landmarks  $\check{\mathbf{x}}_\alpha$  and  $\check{\mathbf{x}}_\beta$  is compared with the Euclidean distance  $r_{ij}$  and relative orientation  $\theta_{ij}$  between two edgels  $\mathbf{x}_i$  and  $\mathbf{x}_j$  that are to be matched to  $\check{\mathbf{x}}_\alpha$  and  $\check{\mathbf{x}}_\beta$ , respectively, and the probability of this pair of matches is expressed as [90]

$$p(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha, \mathbf{x}_j \leftrightarrow \check{\mathbf{x}}_\beta) = \frac{1}{2\pi\sigma_r\sigma_\theta} e^{-\frac{(r_{ij}-r_{\alpha\beta})^2}{2\sigma_r^2} - \frac{(\theta_{ij}-\theta_{\alpha\beta})^2}{2\sigma_\theta^2}} \quad (3.58)$$

where  $\sigma_r$  and  $\sigma_\theta$  are some appropriately chosen constants,  $-90 \leq \theta_{\alpha\beta} \leq 90$  and  $-90 \leq \theta_{ij} \leq 90$ .

Selected values for  $\sigma_r$  and  $\sigma_\theta$  are dependent on the image size and the image quality. For our experiments, we estimate these values according to the scaling factor and



**Figure 3.15:** Result of the probabilities (a) Initialised one. (b) After 5 iterations. (c) After 10 iterations. (d) After 30 iterations. (e) After 70 iterations. (f) Final one (iteration=300).

**Table 3.1: Number of bad matches for different values of  $\sigma_\theta$** 

$\sigma_\theta$	10	20	40
number of bad matches	6	5	9

the local window we place around each landmark. These values should not be too small or too large. We define the value of  $\sigma_r$  to be equal to  $5 \times s$  ( $s$  is the scaling factor). This is because we expect for  $s < 1$ , the image to be smaller than the model, and so the detail in the image not to be very high in comparison with a case when  $s > 1$ , i.e. when the image has much higher resolution than the model. We consider, therefore, rational to use a grosser histogram representation for the shape context in cases of small images than in cases of large, high resolution images. Then, we tried different values of  $\sigma_\theta$  according to the size of the local window (in this example 41). Table 3.1 shows the results for the image we use for this assessment. The results show that the best value for  $\sigma_\theta$  is 20, which is about equal to half the size of the local window times the scaling factor ( $\approx 20 \times s$ ). For the same reasons discussed above, we used  $\sigma_\theta$  to be proportional to the scaling factor for all other experimental images.

Then, the probabilities can be updated according to

$$P^{(n+1)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha) = \frac{P^n(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha)Q^{(n)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha)}{\sum_\beta P^{(n)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\beta)Q^{(n)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\beta)} \quad (3.59)$$

where

$$Q^{(n)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha) = \prod_j \sum_\beta P^{(n)}(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha) p(\mathbf{x}_i \leftrightarrow \check{\mathbf{x}}_\alpha, \mathbf{x}_j \leftrightarrow \check{\mathbf{x}}_\beta) \quad (3.60)$$

In practice, the algorithm is terminated if any of the following conditions is true:

- If one of the probabilities for each point exceeds  $1 - \epsilon$  ( $\epsilon = 0.1$ ).
- The number of iterations has exceeded some large limit (300).

The results of the probabilities algorithm for different iterations are shown in figure 3.15.

The blue lines are the connecting points lines which are used to show the movement of each point more easily.

After identifying the location of each cephalometric landmark on the input image, the next stage is to find the corresponding curves of the cephalometric landmarks (shape template). The method that is utilized in this stage is the snake method described in chapter 2. As it can be seen from figure 3.2, each cephalometric landmark is located on a curve or a line. At this stage, we first place all curves on the input image according to the final position of the cephalometric landmarks which are considered as initial snakes (see figure 3.16(a)). For the soft tissue profile, the outlier of the mask that we used before is used as its initial position.

The shape of the final snakes must be the same as the template shape. Therefore, in order not to allow the snake to change shape drastically, the snake algorithm is modified by defining a constraint energy term and removing the internal energy term (see equation (2.1)).

We define the constraint energy as

$$E_{constraint} = \lambda \sum_{k=1; k \neq i}^n ((x_i - x_k - d_{ik;x})^2 + (y_i - y_k - d_{ik;y})^2) \quad (3.61)$$

where  $\lambda$  is a constant ( $\lambda \leq 1$ ),  $n$  is the total number of snake points,  $d_{ik;x}$  and  $d_{ik;y}$  are the distances of snake point  $i$  from snake point  $k$  along the  $x$  and  $y$  direction, at time ( $t = 0$ ), respectively. In other words, they are the distances of the initial snake point  $i$  from the initial snake point  $k$ , and we wish to preserve as much as possible the initial relative position of the points that make up the snake.

Then, the forces that correspond to the constraint energy term are

$$F_{x(constraint(i))} = \frac{\partial E_{constraint}}{\partial x_i} = \lambda \sum_{k \neq i} 2(x_i - x_k - d_{ik;x}) = 2\lambda((n-1)x_i - \sum_{k \neq i} (x_k + d_{ik;x})) \quad (3.62)$$

$$F_{y(\text{constraint}(i))} = \frac{\partial E_{\text{constraint}}}{\partial x_i} = \lambda \sum_{k \neq i} 2(y_i - y_k - d_{ik;y}) = 2\lambda((n-1)y_i - \sum_{k \neq i} (y_k + d_{ik;y})) \quad (3.63)$$

Each of the above two equations represents a set of linear equations. These equations may be written in matrix form as

$$F_{x(\text{constraint})} = CX + CX_0 \quad (3.64)$$

$$F_{y(\text{constraint})} = CY + CY_0 \quad (3.65)$$

where

$$C = \begin{bmatrix} 2\lambda(n-1) & -2\lambda & -2\lambda & \cdots & -2\lambda & -2\lambda \\ -2\lambda & 2\lambda(n-1) & -2\lambda & -2\lambda & \cdots & -2\lambda \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -2\lambda & -2\lambda & -2\lambda & \cdots & 2\lambda(n-1) & -2\lambda \\ -2\lambda & -2\lambda & -2\lambda & \cdots & -2\lambda & 2\lambda(n-1) \end{bmatrix} \quad (3.66)$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (3.67)$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (3.68)$$

The snake forces can be written as

$$F_{x(\text{snake}(i))} = F_{x(\text{constraint}(i))} + F_{x(\text{image}(i))} \quad (3.69)$$

$$F_{y(\text{snake}(i))} = F_{y(\text{constraint}(i))} + F_{y(\text{image}(i))} \quad (3.70)$$

Note that  $F_{x(image(i))}$  and  $F_{y(image(i))}$  are explained in detail in chapter 2.

Equation (3.69) and (3.70) may be rewritten in matrix form as (see also equation (2.52) to (2.61))

$$CX_t + CX_0 + F_{x(image)}(X_{t-1}, Y_{t-1}) = -\gamma(X_t - X_{t-1}) \quad (3.71)$$

$$CY_t + CY_0 + F_{y(image)}(X_{t-1}, Y_{t-1}) = -\gamma(Y_t - Y_{t-1}) \quad (3.72)$$

Then, the snake position at time  $t$  is given by

$$X_t = (C + \gamma I)^{-1}(\gamma X_{t-1} - CX_0 - F_{x(image)}(X_{t-1}, Y_{t-1})) \quad (3.73)$$

$$Y_t = (C + \gamma I)^{-1}(\gamma Y_{t-1} - CY_0 - F_{y(image)}(X_{t-1}, Y_{t-1})) \quad (3.74)$$

If  $\lambda = 1$ , then the final snake has the same shape as the initial snake (it does not let the snake to deform). If  $\lambda$  is chosen to be very small, then the final snake may not have the same shape as the initial one. Therefore, the value of  $\lambda$  controls the deformation of the snake. From our experiments, we concluded that a good value of  $\lambda$  is 0.01. However, acceptable results are obtained with  $\lambda$  in the range  $[0.001, 0.3]$ . In addition, the values of other weighting coefficients are equal to 1 ( $w_{line}$  and  $w_{edge}$ ).

Note that the local histogram equalization image (see chapter 2) is used in the snake method to have a better result (see figure 3.16(b)). Figure 3.16(c) and 3.16(d) shows the result of the snake with  $\lambda = 0.01$ .

The results of applying the algorithm on more X-ray images with different sizes and qualities are shown in figures 3.17, 3.18 and 3.19.

To find the error associated with the identification of the landmarks, first the cephalometric points (see figure 3.2) must be identified on the X-ray image manually. As it can be seen in figure 3.2, the cephalometric point  $S$  is not located on any curve. There-

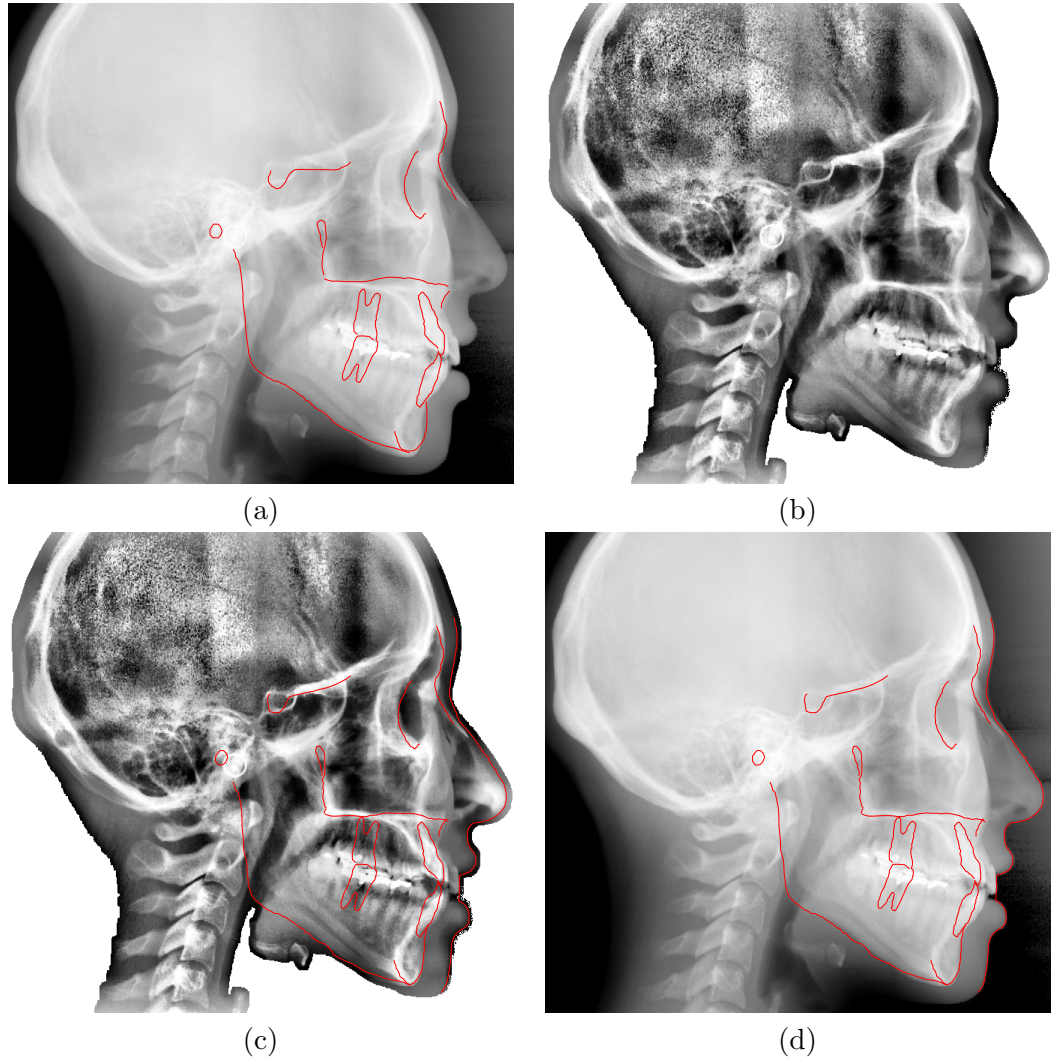
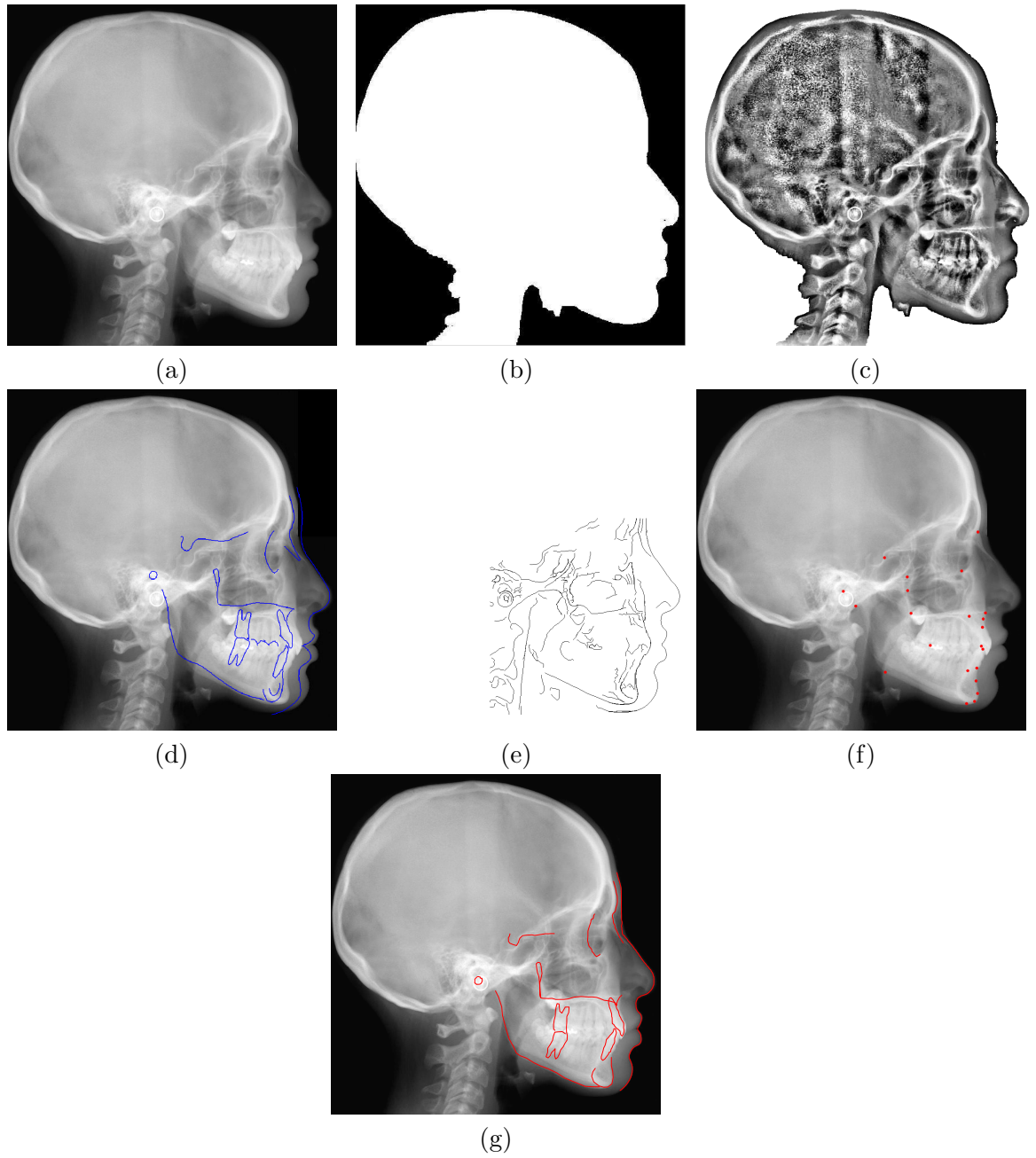


Figure 3.16: (a) Initial position of the snake on the X-ray image. (b) Initial position of the snake on the local histogram equalization of the X-ray image. (c) Final position of the snake on the local histogram equalization of the X-ray image. (d) Final position of the snake on the X-ray image.



**Figure 3.17:** (a) An input image. (b) Mask. (c) Local histogram equalization. (d) Initial cephalometric landmarks on the input image. (e) Edge detection result. (f) Final Probabilistic relaxation result. (g) Snake result.

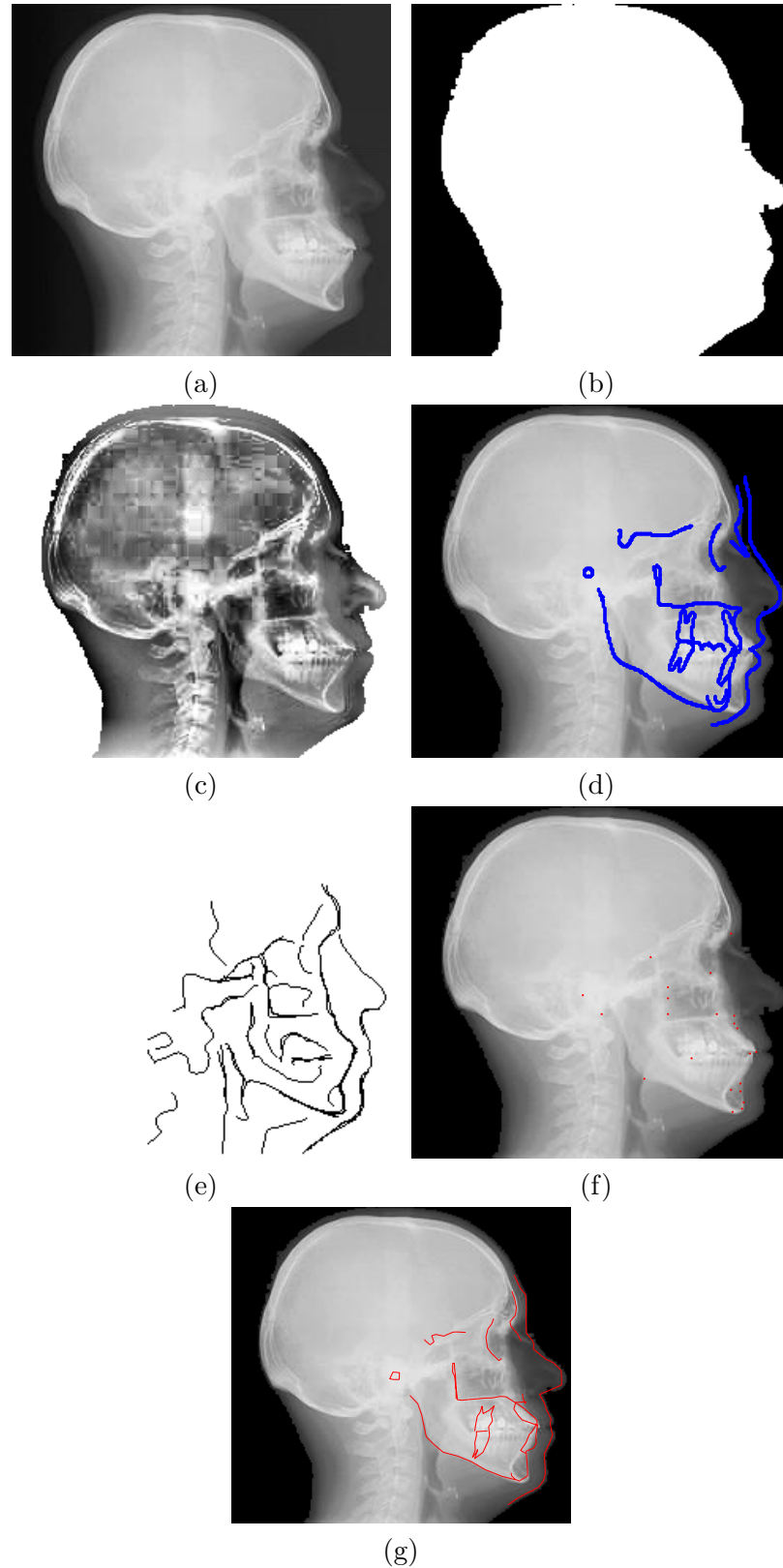
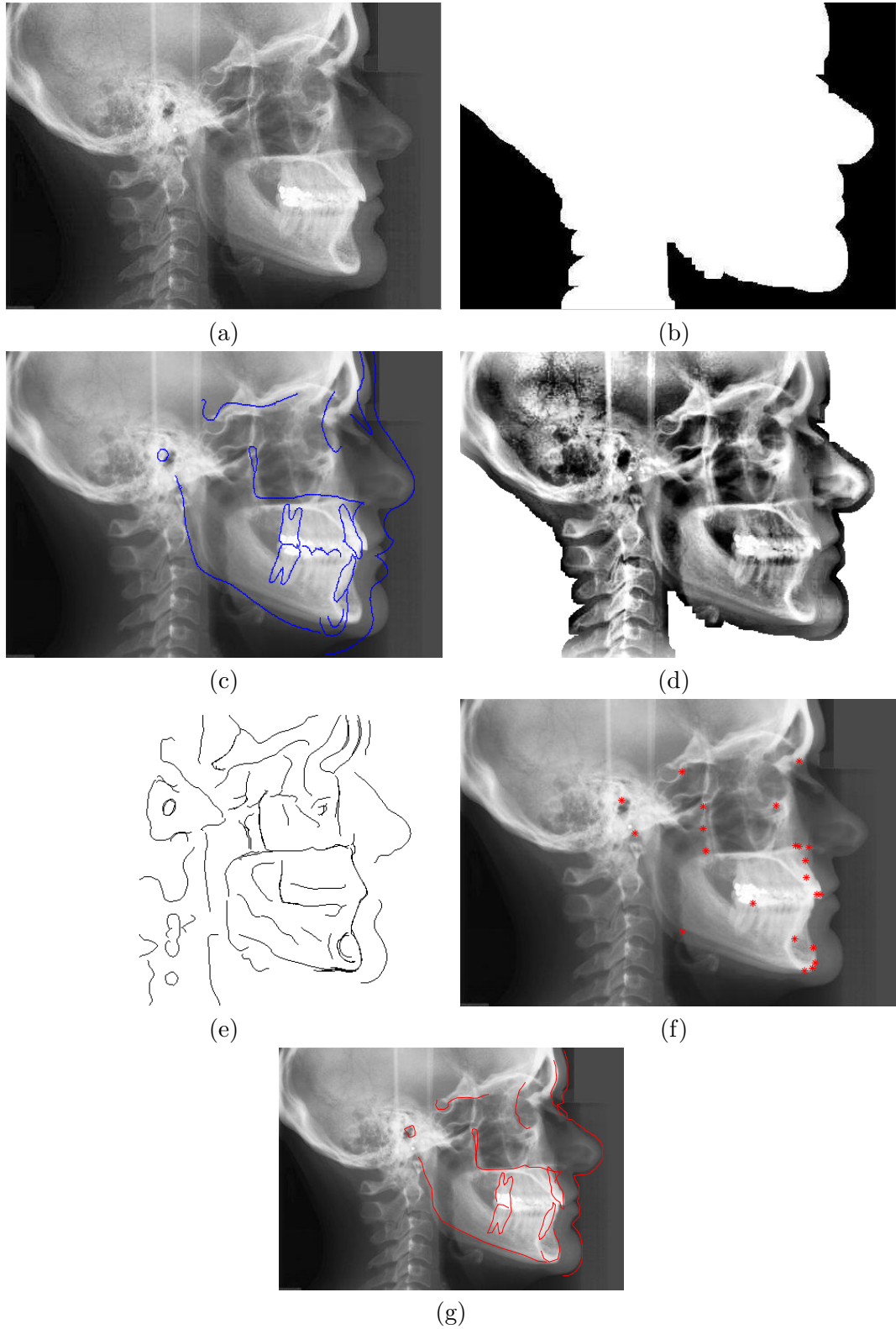


Figure 3.18: (a) An input image. (b) Mask. (c) Local histogram equalization. (d) Initial cephalometric landmarks on the input image. (e) Edge detection result. (f) Final Probabilistic relaxation result. (g) Snake result.



**Figure 3.19:** (a) An input image. (b) Mask. (c) Local histogram equalization. (d) Initial cephalometric landmarks on the input image. (e) Edge detection result. (f) Final Probabilistic relaxation result. (g) Snake result.

**Table 3.2: Experminet results**

Cephalometric points	Accurate $\leq 1$ mm	Accurate $\leq 2$ mm
S	25%	50%
N	50%	100%
ORB	25%	50%
PTMS	25%	50%
PTM	25%	75%
PNS	25%	75%
POR	—	50%
ART	—	75%
GO	25%	75%
ANS	25%	75%
PR	—	50%
A	50%	75%
LIE	25%	75%
UIE	25%	50%
ID	25%	75%
B	25%	75%
POG	25%	100%
GN	50%	100%
ME	50%	100%
OC1	—	25%
UIA	25%	50%
LIA	50%	100%
SOFT TISSUE	50%	75%

fore, instead, two more points marks manually on the curve nearby. Figure 3.20 shows the manually identified cephalometric landmarks (key points) plus a few extra points for the soft tissue profile on the X-ray images. The distances between cephalometric points obtained by hand and caphalometric landmark obtained by the proposed algorithm are found. For point *S* and the soft tissue profile, the distances between cephalometric points obtained by hand and caphalometric landmark curves obtained by the proposed algorithm are found.

Table 3.2 shows the percentage of landmarks identified within 1 mm and 2 mm distance from their true positions. It can be seen that the cephalometric landmarks N, B, ME, LIA and GN on all images are found within the acceptable range. Also, almost 31% of landmarks are identified within 1mm and 71% within 2mm (without considering *S* and the soft tissue).

### 3.6.4 Discussion and conclusions

The results show that the proposed algorithm to identify the cephalometric landmarks is quite successful and on par with other algorithm proposed for the same problem, e.g. they are much better than the results reported in [78] and similar with the results reported

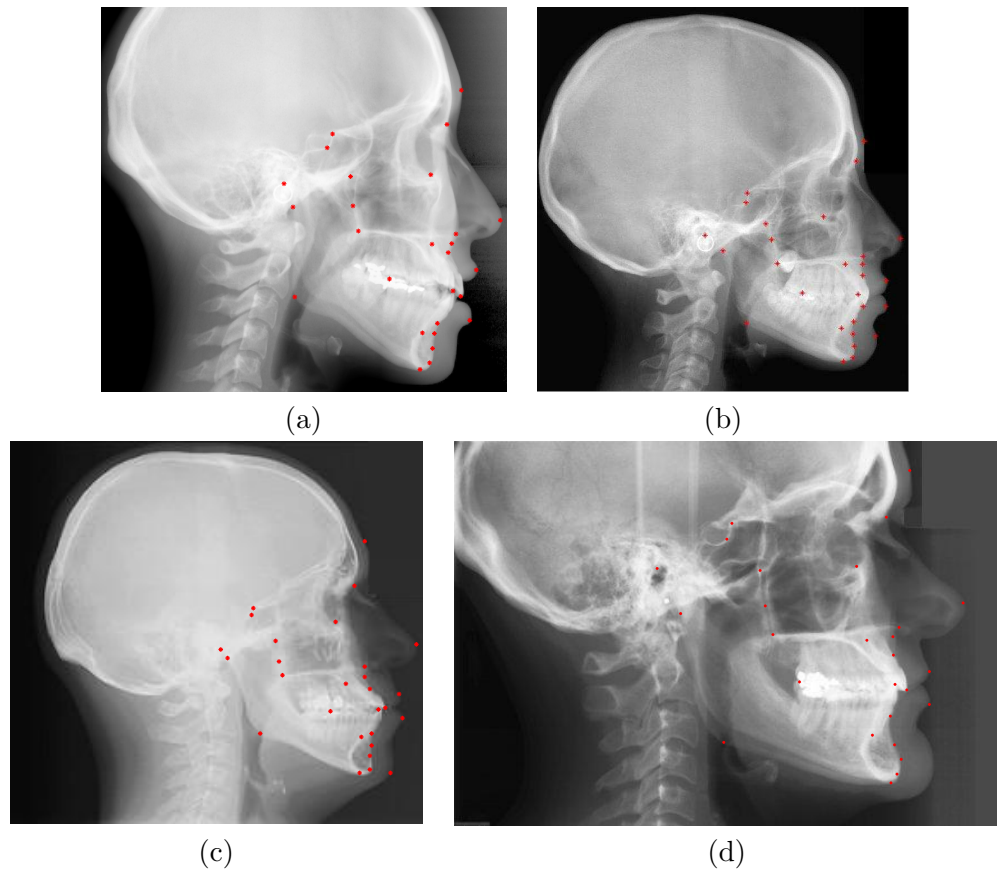


Figure 3.20: a) Cephalometric landmarks points (see figure 3.2) identified by hand on (a) the X-ray image of figure 3.3(b). (b) the X-ray image of figure 3.17(a). (c) the X-ray image of figure 3.18(a). (d) the X-ray image of figure 3.19(a).

in [73] discussed in section 3.3. Especially, it can be a useful method if its results are considered as an initial identification of the cephalometric landmarks as a time saving first step in the manual identification of landmarks. Then, the results of the algorithm may be adjusted by an expert. Also, the algorithm may be improved if the characteristics of each specific landmark are also considered separately. For instance, by placing a window around each identified landmark and trying to adjust the point based on surrounding grey level values.

### 3.7 Conclusions from this chapter

This chapter described a new algorithm for identifying the cephalometric landmarks on the X-ray images, which combines contextual information from the general shape of the bones of the head (used as measurements specific to each landmark in the form of its shape context) and relational information expressing the relative position of the landmarks with respect to each other. After providing some literature survey in cephalometric landmark identification methods, the probabilistic relaxation algorithm proposed by Christmas et al. [90] was described in detail. Then, the new algorithm based on the Christmas algorithm and the shape context method was described and implemented. From the results of the algorithm, it was concluded that the proposed algorithm to automatically identify cephalometric landmarks produces results that are comparable in quality with results reported in the literature by most other algorithms. It seems that the identification of the cephalometric landmarks in digital images has reached a point of accuracy that cannot easily be broken, most likely due to the poor quality of the images themselves captured for routine clinical studies, their digitisation and possibly poor compression process to which they might have been subjected.

## Chapter 4

# Asymmetry Identification Using Generalised Symmetry Axis

### 4.1 Introduction

COMPUTER TOMOGRAPHY (CT) images are X-ray based and they are not only used for diagnostic purposes, but also for treatment and surgical planning, and for determining the prognosis for various conditions. In CT images, one may distinguish between different types of tissue, such as bone, muscle and fat. Facial and skull CT scanning is often used when fast surgical diagnosis and planning are vital. The vast majority of facial and head trauma patients with isolated or severe injuries are injured as a result of road traffic accidents, falling, or beating. Internal bleeding, injuries and deformities in these kinds of patient destroy the symmetry, for example, of the frontal view of the face. Although there are many works on finding the reflective symmetry axis in the fields of mathematics, computer vision and image processing, the main problem in many natural objects is that the axis of symmetry is not usually just a straight line, but rather a generalised curve. In this chapter, we develop methodology for identifying the breaking of symmetry in CT images of parts of the human body, that are expected to be symmetric, and quantifying the degree of asymmetry, as a diagnostic support tool for surgeons. To quantify asymmetry, first, the symmetry axis between the two halves of the

depicted part has to be defined. This may not be a straight line. We refer to such an axis of symmetry as “generalised axis of symmetry”. Once the generalised symmetry axis is defined, the differences between the two halves of the depicted part are quantified. Both these stages are fully automated.

This chapter first gives an introduction to symmetry and then it specifically focuses on methods of determining an axis of local symmetry.

## 4.2 Symmetry

Symmetry is one of the most important features of shapes and objects which can be found in many man-made and natural scenes. The most widely used definition of symmetry was given by mathematician and physicist Weyl in 1952 [93]. He defined an object as being a symmetric object if, when a certain operation is applied to it, it appears exactly the same after the operation. Many techniques of detecting symmetry treat symmetry as a binary feature. This means that either symmetry exists or not. However, it should be considered that perfect symmetry only exists in the mathematics world. Therefore, symmetry may be also seen as a continuous feature which expresses the amount of deviation of a given shape from perfect symmetry [94] [95].

Symmetry detection methods can broadly be divided into two sets: global feature methods and local feature methods. In global feature techniques, the whole object contributes to determining the symmetry (e.g. [96] [97] [98] [99] [100]), while in local feature techniques, only parts of an object or subsets of its points are symmetric with respect to a given symmetry (e.g. [101] [84] [102] [103] [104] [105] [106] [107]). The advantage of local feature methods is to detect more efficiently local symmetries in images that are not globally symmetric.

## 4.3 Types of symmetry

The two most studied types of symmetry are the following.

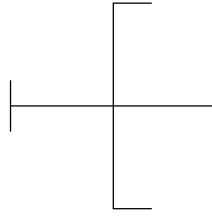


Figure 4.1: Reflectional Symmetry

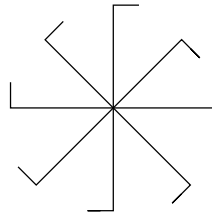
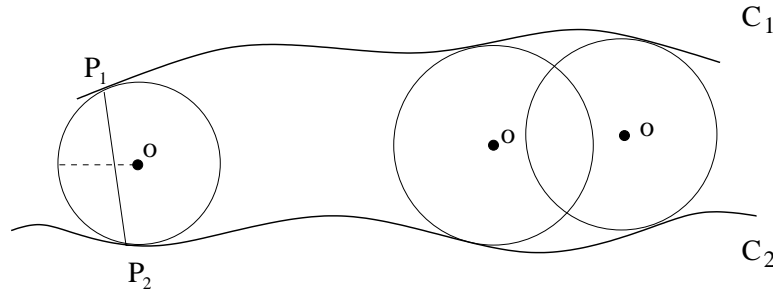


Figure 4.2: Rotational Symmetry

1. Bilateral symmetry (also called mirror symmetry or reflectional symmetry): an object is said to possess mirror symmetry if it is invariant under a reflection about a line (axis of symmetry) in  $2D$  or a plane in  $3D$  (see figure 4.1).
2. Rotational symmetry: an object is said to possess rotational symmetry of order  $n$  if it is invariant under rotation of  $\frac{2\pi}{n}$  radians about the center of mass of the object in  $2D$  or a line passing through the center of mass of the object in  $3D$  (see figure 4.2).

## 4.4 Previous work on symmetry detection

Symmetry detection and analysis has been used in various applications in computer vision, ranging from facial image analysis [108] and vehicle detection [109], to medical image analysis [110] [111] and texture discrimination [112] [113]. In various neurological and plastic surgery applications, the symmetry between two halves of the imaged organ has to be assessed and quantified. There are several researchers who have worked on detecting reflective symmetry in medical images. Alterson and Plewes [114] presented a symmetry detection algorithm for breast cancer determination based on feature extrac-



**Figure 4.3:** The SAT symmetry axis is the locus of the circle centers  $O$ .

tion techniques. Junck et al. [115] developed a method for automatic detection of line symmetry in brain images based on correlation analysis. Several more algorithms have been proposed to detect the symmetry plane in brain images that may be used for tumor extraction (e.g. [116] [117] [110]).

As mentioned earlier, the symmetry detection techniques use an entire image as a signal or they use local features such as edges, contours and boundary points.

#### 4.4.1 Symmetry detection from borders

One of the first attempts to detect symmetry starting from the borders of an object is the symmetric axial transform (SAT). The SAT algorithm, also known as medial axis transform (MAT) which is a method to obtain local symmetry axes, was first introduced by Blum in 1967 [118] and improved by him and Nagel later [119] [101]. The method described objects as the selection of maximal discs inside them. Each such disc must touch the border of the object at least in two points. A maximal disk is one that should not contain entirely another disk and must lie within the object. The locus of centers of them was called symmetry axis. The major disadvantage of this method is its high sensitivity to noise on the shape boundary. Several techniques have been proposed based on this algorithm. Figure 4.3 illustrates the SAT method for two curves  $C_1$  and  $C_2$ , where the symmetry axis is the locus of the centers of the circles which are at least bitangent to the contours while at the same time being contained entirely within the contours' interior. In 1984, Brady and Asada [102] attempted to solve some limitations of the SAT algorithm by introducing the smoothed local symmetry (SLS). A pair of points  $P_1$  and  $P_2$  (see figure

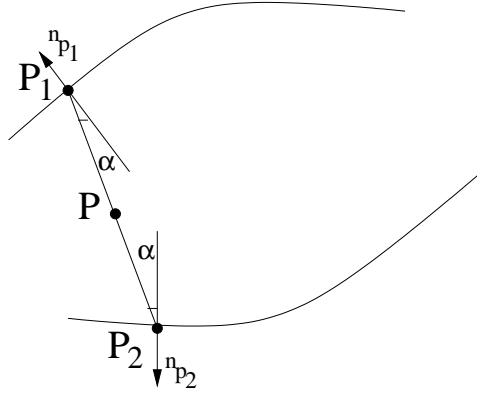
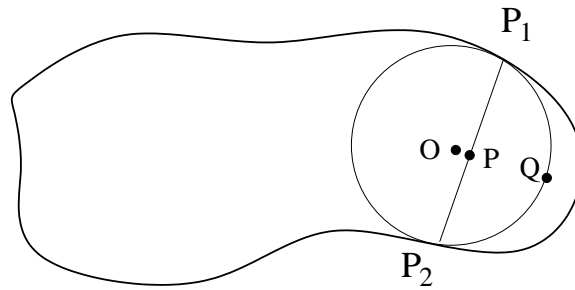


Figure 4.4: Smooth Local Symmetries (SLS)

Figure 4.5: Different methods identify different local symmetry points:  $O$ : SAT,  $P$ : SLS, and  $Q$ : PISA

4.4) on the curves are locally symmetric if the angles between segment  $\overline{P_1P_2}$  and the two normals to the curves at these points are equal. The midpoint  $P$  of a segment connecting these locally symmetric points is a symmetry locus. Therefore, the local axis of symmetry is represented by the locus of the midpoints of the line segments joining symmetrical points.

Process-Infering Symmetry Analysis (PISA) is another symmetry detection method which was proposed by Leyton in 1988 [120]. It defines the axis of symmetry as the trajectory of the midpoints of the arc of the moving circle. The geometry of different symmetry points defined by different methods was discussed by Cho and Dunn [107]. Figure 4.5 shows the geometry of some of the different local symmetry points identified by the methods of SAT, SLS and PISA. Mukherjee et al. [121] introduced a method based on perspective or affine transformation which was successfully used to find symmetries on

contours of simple shapes. To form symmetry hypotheses, they applied a matching based on affine semi-local invariants. Partial occlusion is another method which was proposed by Sato and Cipolla in 1997.

Several researchers worked on detecting axes of symmetry from borders such as Atallah in 1985 [122], Sato and Cipolla in 1997 [123], and Bruckstein and Shaked in 1998 [124].

#### 4.4.2 Symmetry detection from image intensity

In the above described methods, the symmetries of shapes are defined by their boundaries' contours. There are a number of methods that used the intensity of images to detect symmetries. One of the first attempts to use grey level information was investigated by Marola [97] in 1989. His method was based on both intensity images and (digitized) planar curves. Gofman and Kiryati [125] introduced an algorithm from Marola's one based on the Radon and the Fourier transforms. Their algorithm attempted to detect symmetries in an image without using any segmentation procedure.

In 1996, Di Gesu and Valenti [126] proposed another method to measure image symmetry called the Discrete Symmetry Transform (DST). DST is based on the calculation of local axial moments of a body around its center of gravity. In the case of images, intensities are considered as point masses. A year later, in 1997, they [127] introduced the pyramid version of the DST which allows the detection of local symmetries at different resolution levels.

In 1997, Manmatha and Sawhney [128] introduced a local symmetry measure method to detect significant structures in images. Their method was based on the first and second derivatives of Gaussians.

Shen et al. [129] proposed a method to detect global reflectional axes. Their method consists of two energy terms: symmetric energy term, which corresponds to the symmetric components of an object, and asymmetric energy term, which corresponds to asymmetric components. In their method, all the symmetry axes of an object can be detected from the local minima of the asymmetric energy term which is defined as a summation of a set

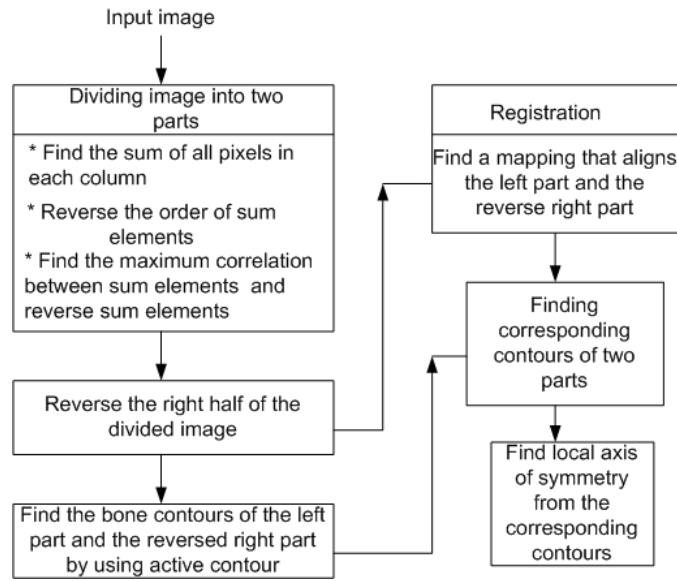


Figure 4.6: Overview of the proposed method

of generalised complex (GC) moments computed for the object. They claimed that their method is robust against both noise and slight deformation by showing the results of the algorithm on typical logo images and human brain images.

## 4.5 The proposed method

The proposed method combines registration and the local axis of symmetry from extracted bone contours together to generate the desired symmetry axis. The method consists of three independent stages:

- finding grossly an axis that allows the division of the image into two parts, loosely called “half” images, but with the understanding that they are not exactly the two halves of the image but rather the two halves of the depicted object;
- finding a mapping that aligns the first half of the image with the second half;
- finding local axes of symmetry of bones from their contours.

The diagram in figure 4.6 gives an overview of the whole process.

### 4.5.1 Dividing the image into two grossly symmetric parts

Let  $I(x, y)$  be an input image of size  $M \times N$ . To detect a suitable axis for dividing the image into two parts, roughly symmetrical to each other with respect to the axis, the correlation method is used in the following way. Note that the tacit assumption here is that the axis of symmetry is roughly along the vertical direction.

- Find the sum of all pixels in each column of the image

$$S_1(k) = \sum_{i=1}^M I(i, k), \quad 1 \leq k \leq N \quad (4.1)$$

- Form a digital signal  $\tilde{S}_1 \equiv (S_1(1), S_1(2), \dots, S_1(N))$ . Reverse the order of elements of  $\tilde{S}_1$  to form digital signal  $\tilde{S}_2$ :

$$\tilde{S}_2 \equiv (S_1(N), S_1(N-1), \dots, S_1(1)) \quad (4.2)$$

- Find the correlation between  $\tilde{S}_1$  and  $\tilde{S}_2$ , assuming that the two signals are repeated ad infinitum with period  $N$

$$c(k) = \sum_{i=1}^N \tilde{S}_1(i+k) \tilde{S}_2(i), \quad k = 1, 2, \dots, N \quad (4.3)$$

with the understanding that  $\tilde{S}_1(i+k) = \tilde{S}_1(\text{mod}_N(i+k))$

- The shift  $k^*$  that corresponds to the maximum correlation ( $\max\{c(k), 1 \leq k \leq N\}$ ) identifies the column that divides the image into two grossly symmetric parts.

### 4.5.2 Registration

Image registration is a technique of finding a geometrical transformation or mapping which aligns two or more images, from single or from different modalities, according to some similarity measure. Then, by using this transformation, a correspondence between pixels in these images is established. The goal of using image registration in this project is to find a transformation that aligns the left part of an input image with the right part. After,

the input image has been divided into two parts, the right part is left to right mirrored, to facilitate the registration process. We refer to these two half images as two separate images denoted by  $I_1(x, y)$  and  $I_2(x, y)$ . To find a correspondence function  $T$ , that takes each point  $(x, y)$  in image  $I_1(x, y)$ , which is called the “source” image, and finds the corresponding point  $T(x, y)$  in image  $I_2(x, y)$ , which is called the “target” image, we use an intensity-based registration algorithm assuming an affine transformation [130] [131] [132]. In intensity-based algorithms, the intensity of the images is used to calculate how similar the two images are.

The main key of any registration method is to find the parameters of a geometric transformation ( $T : (x, y) \rightarrow (\hat{x}, \hat{y})$ ) which maps old coordinates  $(x, y)$  to new coordinates  $(\hat{x}, \hat{y}) = f(x, y)$ , where  $f$  is the function of the 2D spatial-coordinate transformation. If function  $f$  is expressed as two functions  $f_x$  and  $f_y$ , the mapping between the images may be expressed as:

$$I_2(x, y) = I_1(f_x(x, y), f_y(x, y)) \quad (4.4)$$

By assuming the existence of brightness constancy, which states that if a small region is moving, the image intensities between images remain unchanged, and if an affine transformation is assumed, equation (4.4) becomes

$$I(x, y, t) = I(a_1x + a_2y + d_x, a_3x + a_4y + d_y, t - 1) \quad (4.5)$$

where,  $a_1, a_2, a_3$  and  $a_4$  form the affine matrix  $A$

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (4.6)$$

and  $d_x$  and  $d_y$  are the translation parameters forming vector  $\vec{d}$ :

$$\vec{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad (4.7)$$

By defining  $a_5 \equiv d_x$  and  $a_6 \equiv d_y$ , equation (4.5) may be rewritten as:

$$I(x, y, t) = I(a_1x + a_2y + a_5, a_3x + a_4y + a_6, t - 1) \quad (4.8)$$

To estimate parameters  $\vec{a} \equiv (a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6)^T$ , the method of minimising the mean squared error is used

$$E(\vec{a}) \equiv \sum_{(x,y) \in \Omega} [I(x, y, t) - I(a_1x + a_2y + a_5, a_3x + a_4y + a_6, t - 1)]^2 \quad (4.9)$$

where  $\Omega$  denotes a spatial region of interest in the image. Parameter vector  $\vec{a}$  is chosen so that  $E(\vec{a})$  is minimal.

To simplify the minimization, this equation may be approximated by using a first order Taylor series expansion

$$E(\vec{a}) \approx \sum_{(x,y) \in \Omega} (I(x, y, t) - [I(x, y, t) + (a_1x + a_2y + a_5 - x)I_x(x, y, t) + (a_3x + a_4y + a_6 - y)I_y(x, y, t) - I_t(x, y, t)])^2 \quad (4.10)$$

where  $I_x(\cdot)$  and  $I_y(\cdot)$  are the spatial derivatives of  $I(\cdot)$  and  $I_t(\cdot)$  is the temporal derivative of  $I(\cdot)$ .

Now, this error equation may be reduced to:

$$E(\vec{a}) \approx \sum_{(x,y) \in \Omega} ([-(a_1x + a_2y + a_5 - x)I_x(x, y, t) - (a_3x + a_4y + a_6 - y)I_y(x, y, t) + I_t(x, y, t)])^2 \quad (4.11)$$

It may also be expressed as:

$$E(\vec{a}) = \sum_{(x,y) \in \Omega} [k - \vec{c}^T \vec{a}]^2 \quad (4.12)$$

where, scalar  $k \equiv xI_x + yI_y + I_t$  and vector  $\vec{c} \equiv (xI_x \ yI_x \ xI_y \ yI_y \ I_x \ I_y)^T$ .

Now the error function may be minimised by differentiating it with respect to  $\vec{a}$

$$\frac{dE(\vec{a})}{d\vec{a}} = \sum_{(x,y) \in \Omega} (-2)\vec{c} [k - \vec{c}^T \vec{a}] \quad (4.13)$$

So, by setting this derivative equal to zero, we have:

$$\vec{a} = \left( \sum_{(x,y) \in \Omega} \vec{c}\vec{c}^T \right)^{-1} \left( \sum_{(x,y) \in \Omega} \vec{c}k \right) \quad (4.14)$$

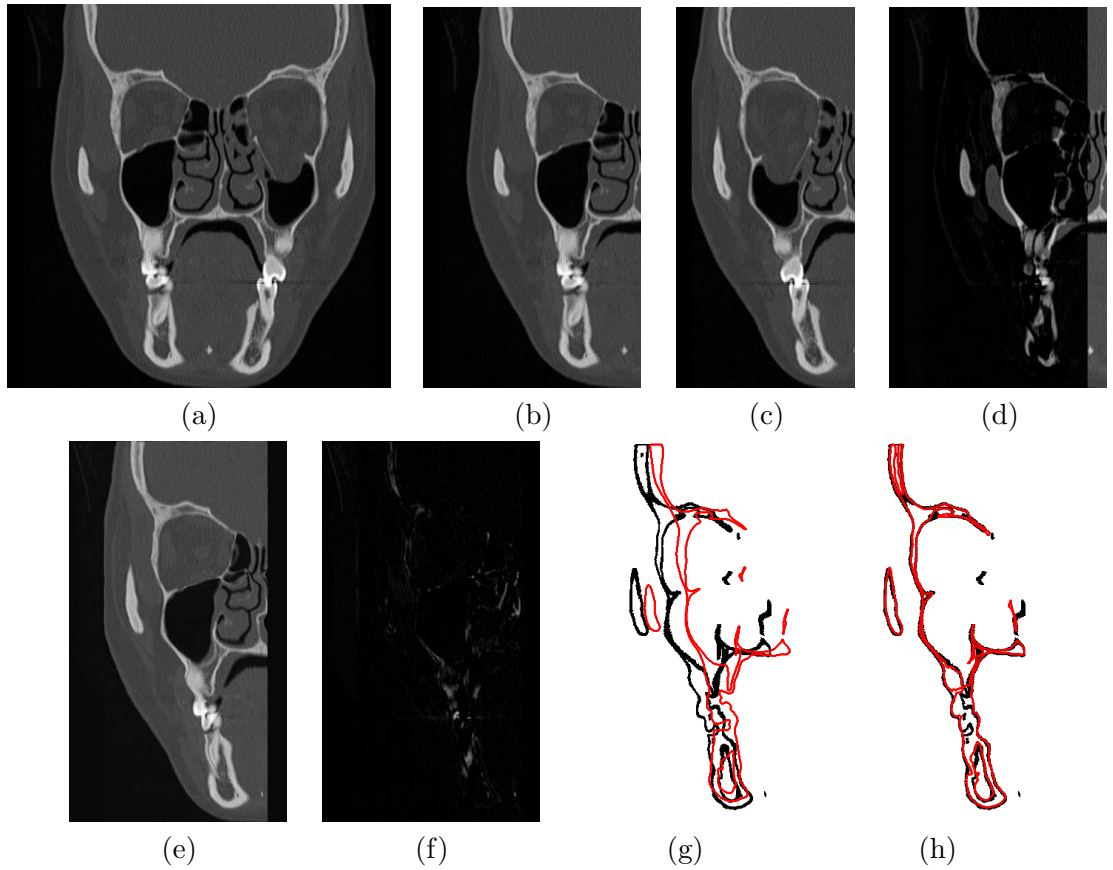
## Results

Figures 4.7 to 4.14 show some results of the registration method. Figures 4.7(b) and 4.7(c) are the left part and the reversed right part of the input image 4.7(a). The difference between these two images (4.7(b) and 4.7(c)) is shown in figure 4.7(d). After applying the registration method and by considering the left part of the input image (4.7(b)) as the source image, and the reversed right part of the input image as the target image (4.7(c)), the transformation parameters were found. Figure 4.7(e) shows the registered source image, and figure 4.7(f) shows the difference between this image and the target one (4.7(c)). Before registration, the level set method had been applied on the input image to segment the bone contours. The contours in the source image were transformed by using the transformation parameters to identify the corresponding contour in the reference image (see figures 4.7(g) and 4.7(h)).

### 4.5.3 Refinement Stage

The next step is to identify the corresponding contours. As our contours are closed curves (or curves that become closed once the image border is considered), we fill each closed contour in each half image with a unique label. Each contour is treated separately, so contours inside contours are represented in different layers as shown in figure 4.15. In this figure,  $l_1, \dots, l_6$  represent distinct labels of the pixels of the corresponding regions.

Once registration of contours has been achieved, we identify for each contour the contour with which it corresponds in the other half of the image, by checking the contour with



**Figure 4.7:** (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

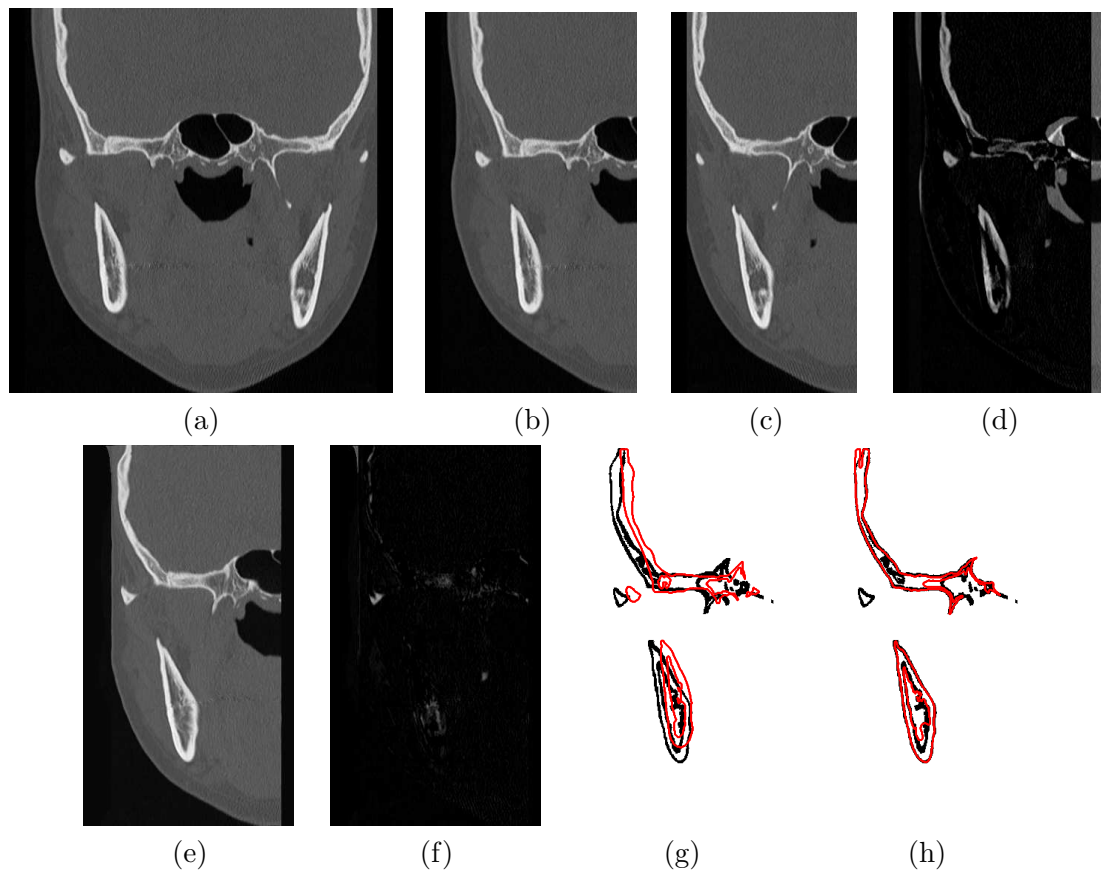


Figure 4.8: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

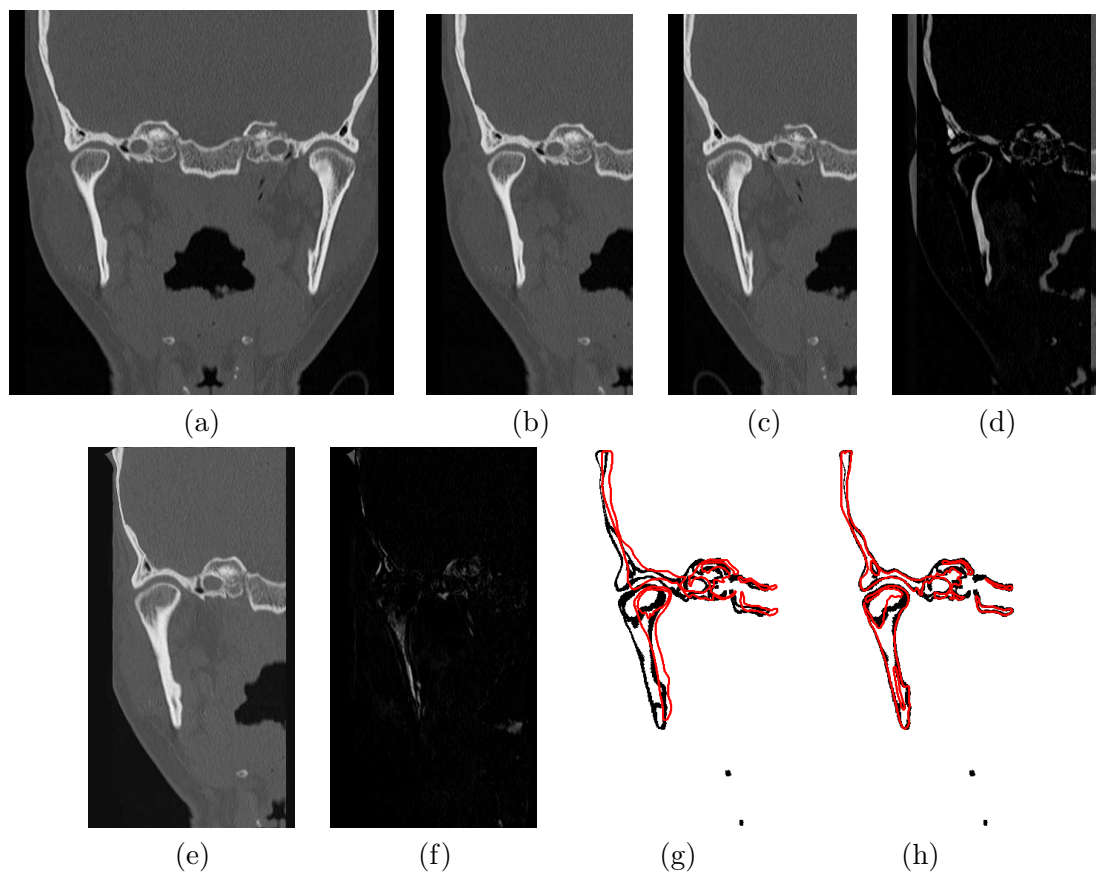


Figure 4.9: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

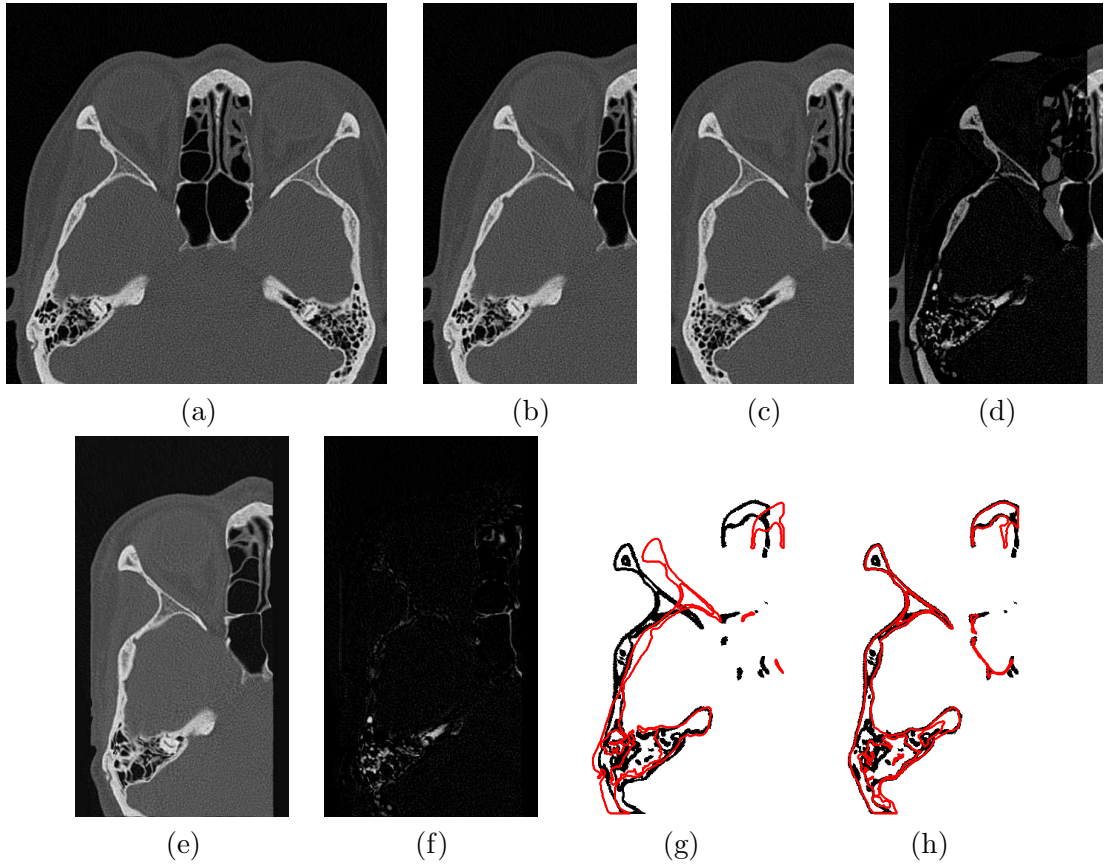


Figure 4.10: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

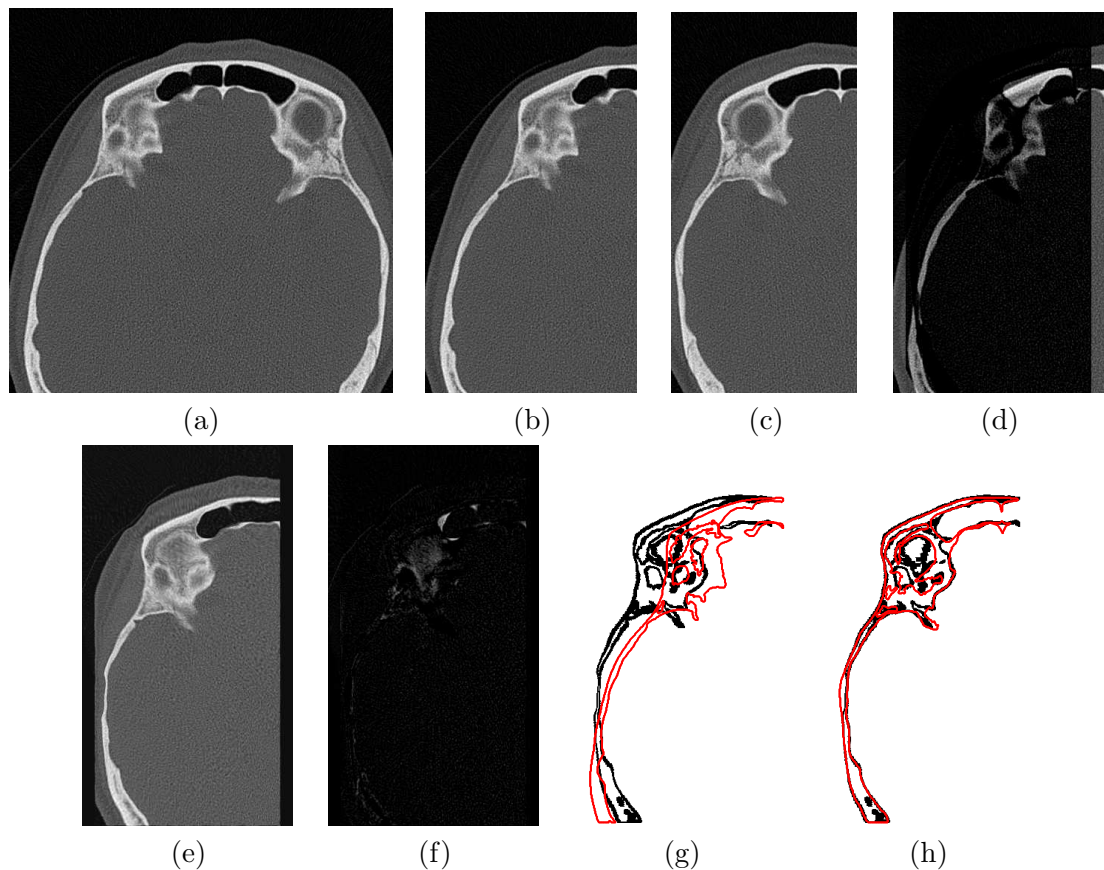


Figure 4.11: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

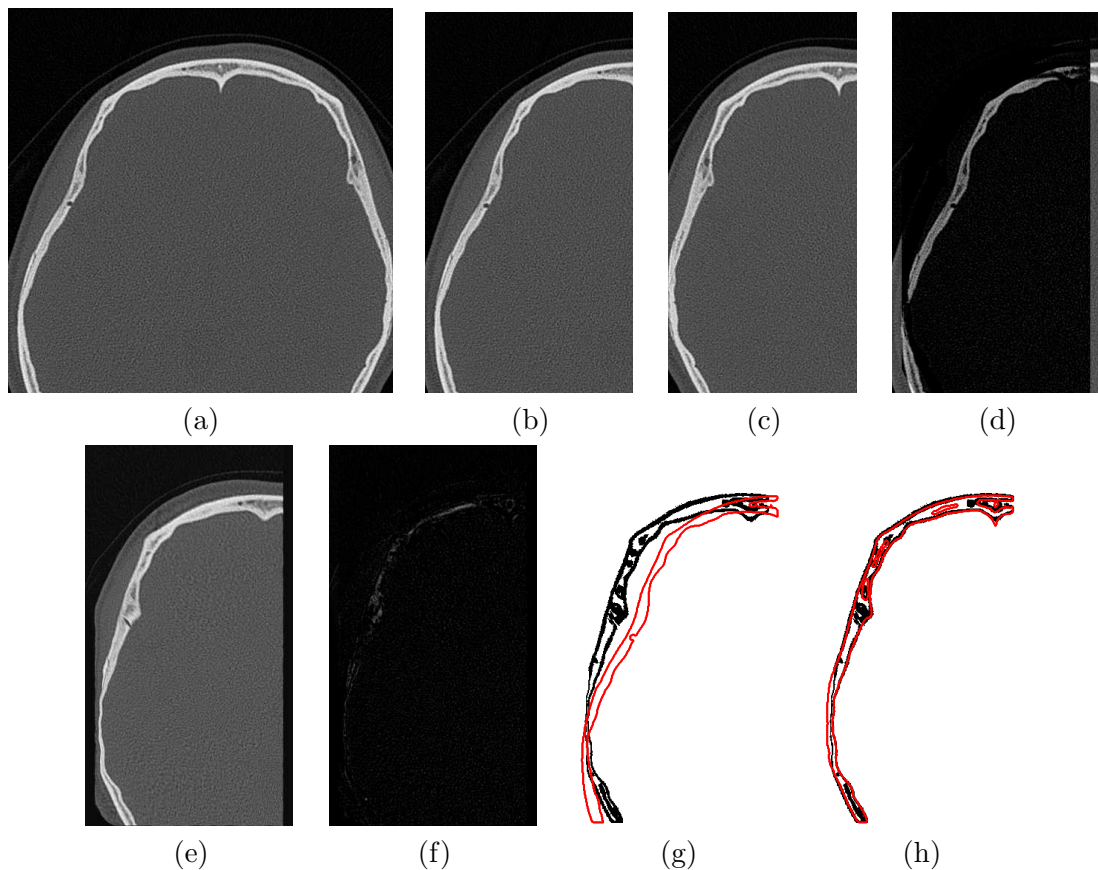


Figure 4.12: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

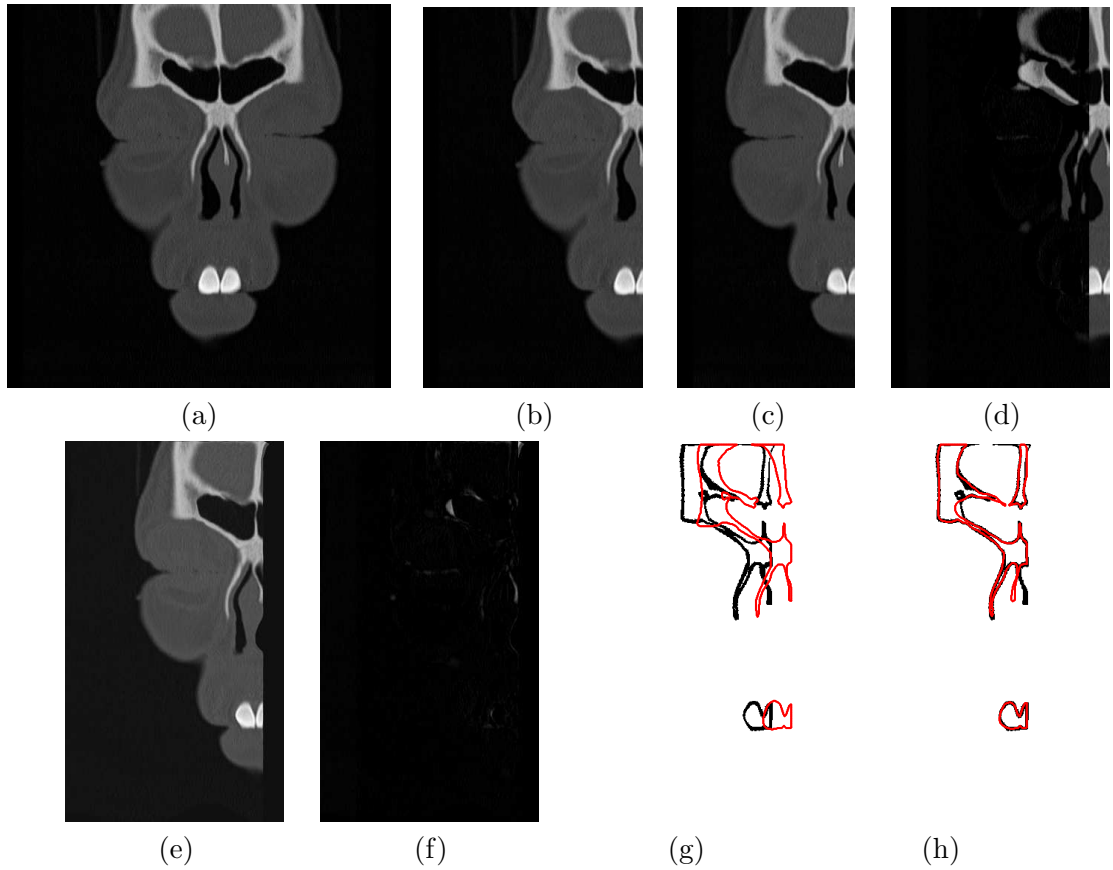


Figure 4.13: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.

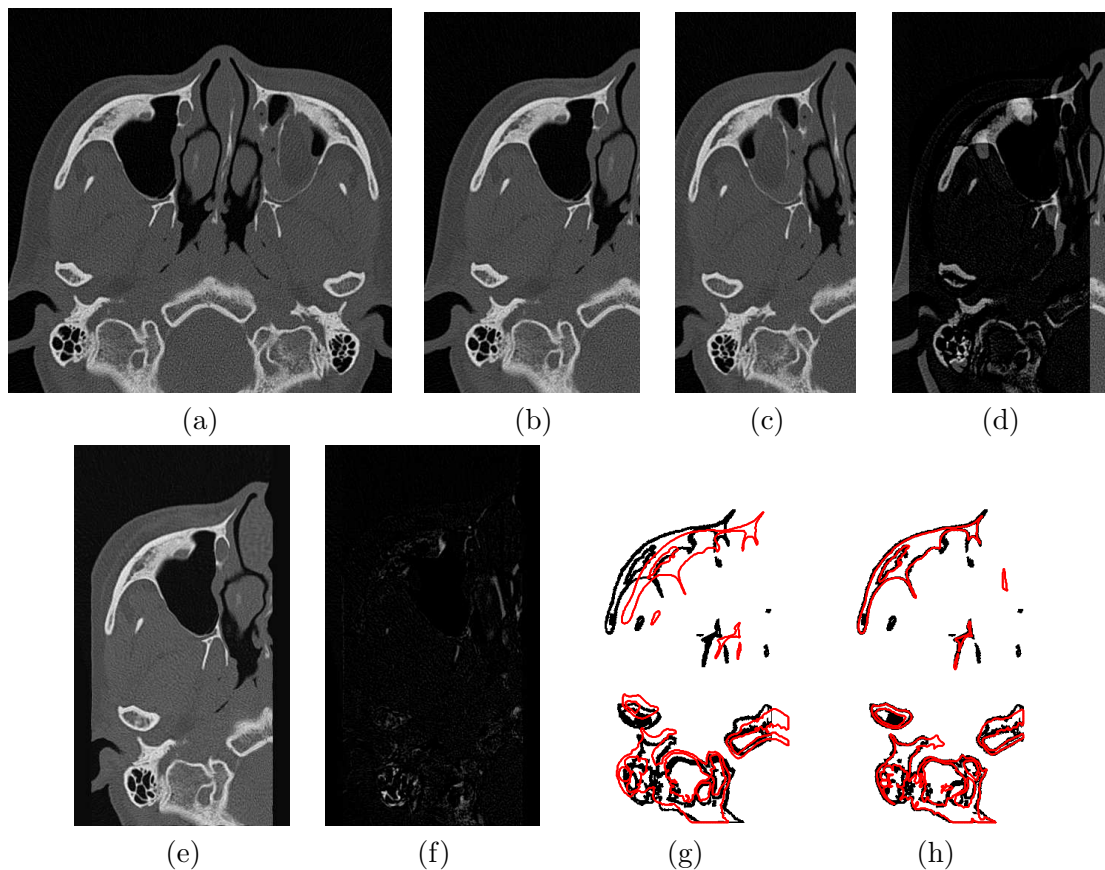
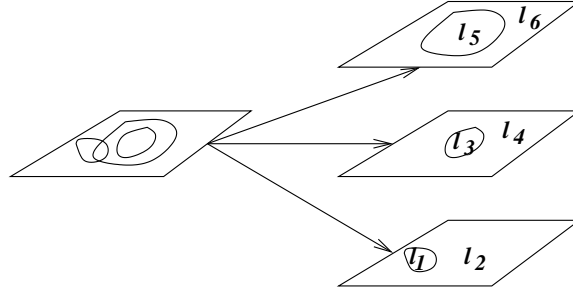


Figure 4.14: (a) Input Image. (b) The left part of the input image. (c) The reversed right part of the input image. (d) The difference between image (b) and (c). (e) Registered source image. (f) The difference between the registered source image and the target image. In the difference images, the absolute value of the difference at each pixel position is used and it is scaled using the same scale for panels (d) and (f), for displaying purposes, between 0 and 255. (g) The black contours are the transformed contours of the original contours of the source image (red ones). (h) The black contours are the transformed source image contours, and the red contours are the target image contours.



**Figure 4.15: Different contours are represented in different layers**

which it shares the maximum pixel overlap. To avoid contours of significantly different sizes being matched, we start by considering the contours with the largest areas.

We start by ranking the contours of both half images according to their area, in decreasing order. We consider the largest contour of one image, and by checking the labels of the registered contours, we identify an overlap measure between this contour and all contours of the other half image with which it has common interior pixels after registration. Let us call  $A$  the contour we consider and  $B_i$  one of the contours of the other half image. For each pair of contours  $A$  and  $B_i$  considered, we compute an overlap measure

$$S(A, B_i) = \frac{\#(A \cap B_i)}{\#(A) + \#(B_i) - \#(A \cap B_i)} \quad (4.15)$$

where  $\#(A)$  means the area i.e. the number of pixels inside contour  $A$ . For contour  $A$  we identify its paired contour  $B_{i^*}$  to be the one that maximises  $S(A, B_i)$ , i.e.  $i^* = \arg\{\max_i S(A, B_i)\}$ . Once two contours have been matched, they are removed from the stacks of contours of the two half images and the next contour in the top of the stack is considered. When one or the other stack becomes empty, the process stops.

The process may be repeated by starting with the contours of the other half-image, in case the order by which we do the matching affects the result. Pairs of contours that have been identified by both routes, are considered as matched contour pairs. Note that this process takes place using the two registered parts of the image, but once a corresponding pair of contours has been identified, the processing that will follow will use the original unregistered contours. This is because the matching we perform at this stage is global

and thus not very accurate, while in the next stage we shall want to define local matching transformations.

After finding the corresponding contours, the next step of this algorithm is to identify the corresponding points on each paired contour. To find which point of the first contour corresponds to which point of the second contour, shape context expressed by using log-polar histograms [92] is used. Consider two paired contours  $A(p)$  and  $B(q)$ , each represented by a set of  $m$  and  $n$  points respectively ( $m \leq n$ ). The shape context of a contour point, is a histogram which expresses the relation of that point to the remaining points of the contour. The histogram is built by dividing the space around that point into  $k$  bins and counting the number of points in each bin. The corresponding points on the two contours are expected to have similar histograms. The shape context of point  $p_i$  of the first contour is defined as

$$h_i(k) = \# \{p \neq p_i : (p - p_i) \in \text{bin}(k), \forall p \in A\} \quad (4.16)$$

where  $\text{bin}(k)$  is uniform in a log-polar coordinate system.

The following steps show how to generate the shape context for each contour.

1) Consider the polar  $(r, \theta)$  coordinate system, where  $r$  and  $\theta$  are radial distance and angle respectively. For each point on the contour, compute the distance  $r$  and angle  $\theta$  to all other points where all angles are between 0 and  $2\pi$ :

$$r = \sqrt{x^2 + y^2} \quad \theta = \arctan \frac{y}{x} \quad (4.17)$$

2) After normalizing the distance by the mean distance, create bins that are uniform in log-polar space.

3) Quantize these distances and angles into the bins (see equation (4.16)). This means that for each point, put all other points in the appropriate bins according to their distances and angles.

After computing  $h_i$  for all points on the two contours, calculate the cost of matching point  $p_i$  on the first contour to a point  $q_j$  on the second contour.

Assume  $C_{ij} = C(p_i, q_j)$  is the cost of matching two points  $p_i$  and  $q_j$  which is defined as

$$C_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (4.18)$$

where  $h_i(k)$  and  $h_j(k)$  are the shape contexts of  $p_i$  and  $q_j$ , respectively.

To find the best match of contour points, the total cost of matching points should be minimised:

$$\min \sum_{i=1}^m C_{i,\pi(i)} \quad (4.19)$$

The final step is to consider the deformation between the actual corresponding contours. For every pair of the matched contours we have thus identified, we calculate now the affine transform that may make them match. The affine transform for a point  $(x, y)$  in one contour and its corresponding point  $(\hat{x}, \hat{y})$  in the other contour, can be defined as

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4.20)$$

Parameters  $a, b, c, d, e$  and  $f$  can be found in the least square error sense based on all the matched points.

#### 4.5.4 Symmetry point identification

We may find a local axis of symmetry between the two contours by computing the center points of the line joining corresponding points of the contours.

#### 4.5.5 Multiple symmetry points

If more than one pair of points are matched along the horizontal direction, we shall extract several points along that direction that could be considered as belonging to the symmetry axis. We present two methods to find the axis of symmetry.

### Averaging method

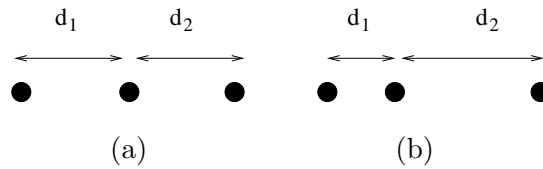
We may use straight averaging to find a single point, or we may use a robust averaging. We distinguish the following cases.

- i) Two points are found. Here we have no other option than to average the two positions.
- ii) Three points are found. Three points may arrange themselves as roughly forming two equal length segments (see figure 4.16(a)) or as in figure 4.16(b).

In case (a) the best we can do is to average the three positions, while in case (b) the best we can do is to ignore the third point as an outlier. We use the following criterion to decide which procedure to follow:

if  $|d_2 - d_1| \geq 2 \min\{d_1, d_2\}$ , average the values of the points that form the segment with  $\min\{d_1, d_2\}$ ; else average all three positions.

- iii) If four or more points are identified as points of the symmetry axis, compute the mean and standard deviation of their positions. Ignore points that are more than one standard deviation away from the mean and recompute a new mean (trimmed mean). The new mean is marked as the position of the local symmetry axis.



**Figure 4.16: Multiple symmetry points**

### 4.5.6 Results

Figures 4.17 and 4.18 show the results of ordering the contours of figures 4.9(b) and 4.9(c) according to their areas from the largest to the smallest, respectively. Figure 4.19 shows the registration result of figure 4.17. Note that after applying the level set method on the full original image, those contours that have area of less than 50 pixels are removed, then each contour is considered as a binary image.

Figure 4.20(a) shows the largest contour of the left half image after registration.

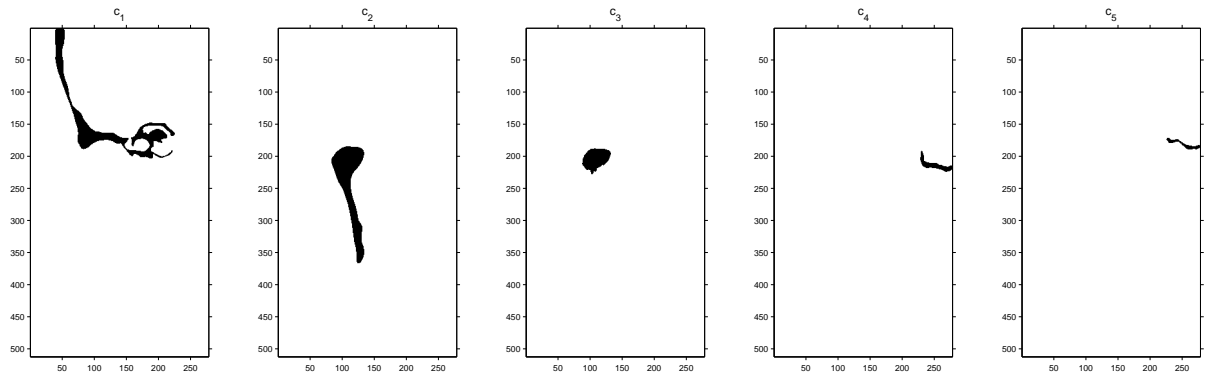


Figure 4.17: Ranking the contours of the left part of the image (see figures 4.9(b)) according to their areas in decreasing order.

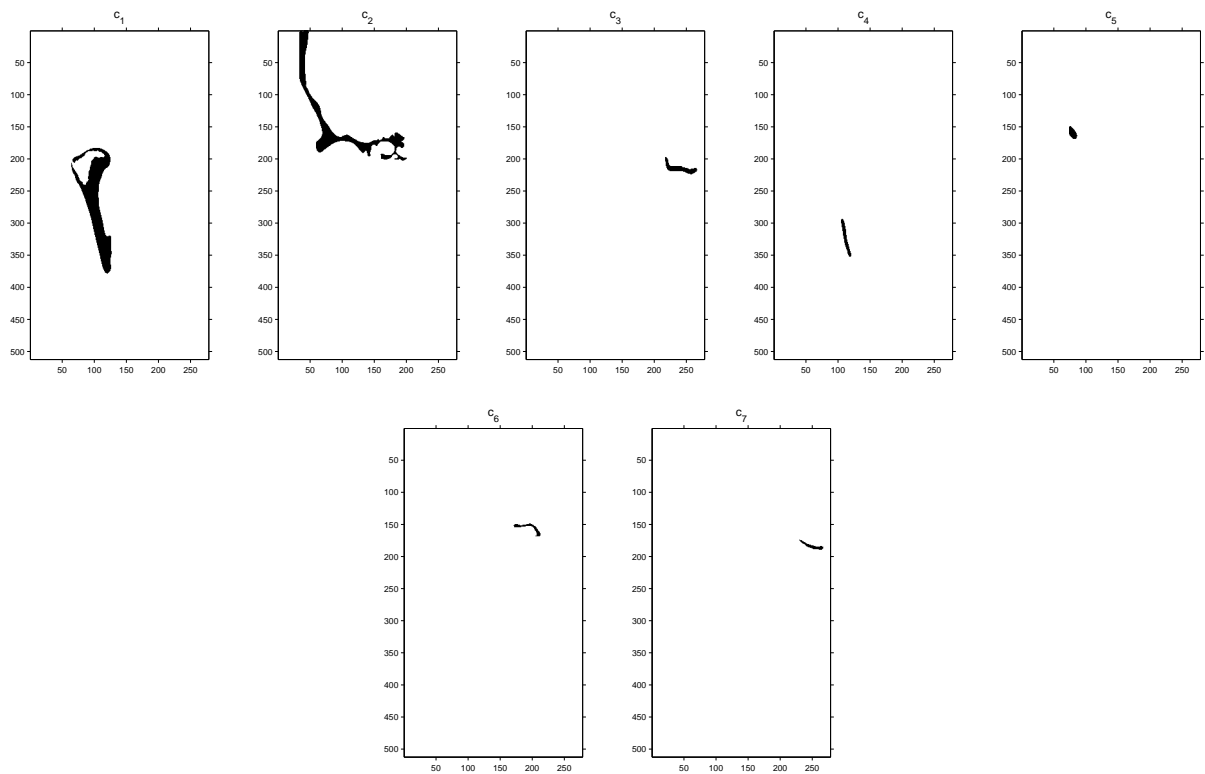
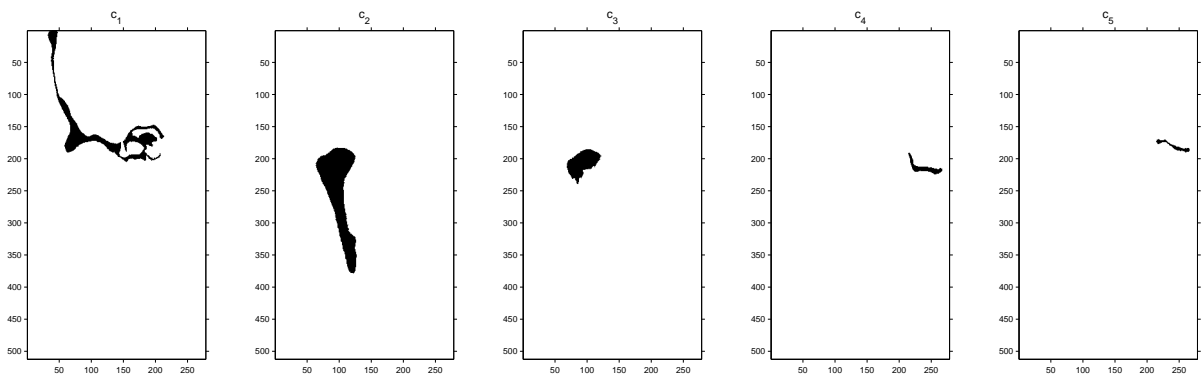


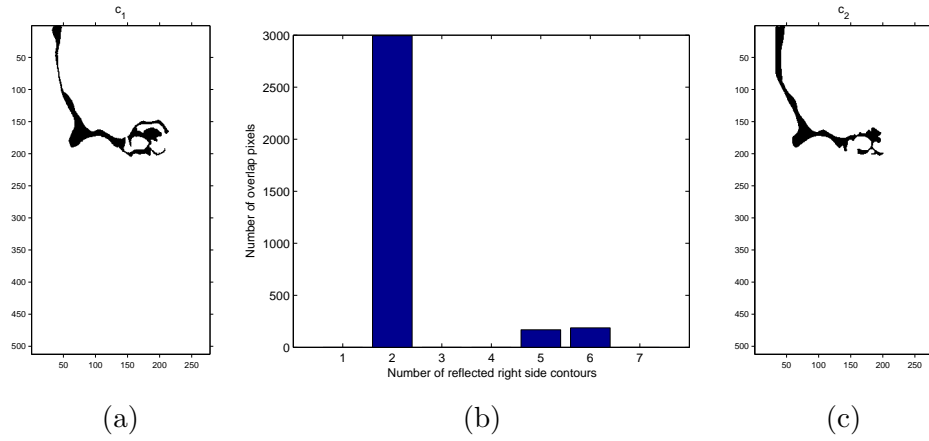
Figure 4.18: Ranking the contours of the reflected right part of the image (see figures 4.9(c)) according to their areas in decreasing order.



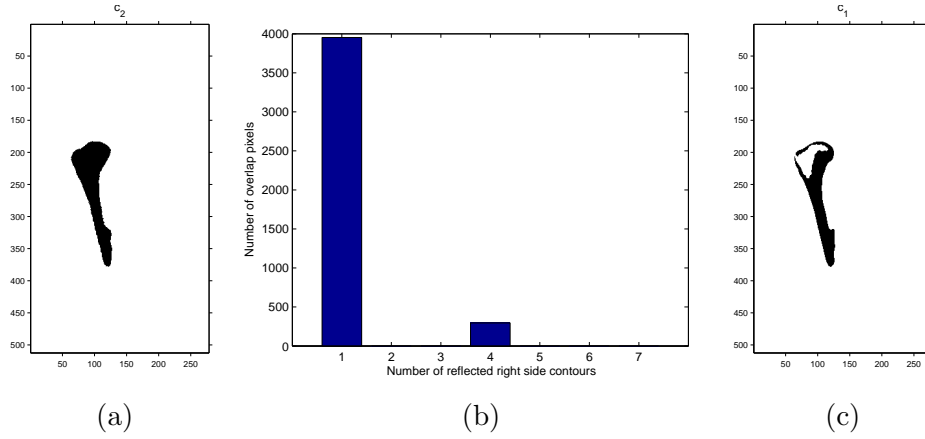
**Figure 4.19:** The registered contours of the left half of the image.

The number of overlapping pixels between this contour and all contours of the reversed right half image can be seen in figure 4.20(b). As it is realized from this graph, the contour has overlapping pixels with the second, fifth and sixth contours of the reversed right half image. However, its corresponding contour is the second one (see figure 4.20(c)) because they have more pixels in common than with the other two.

After this pair from the stacks of the contours was removed, it was attempted to find the corresponding contour of the second largest contour of the left half image (or the largest contour in the stack). Figure 4.21(b) shows which contours of the right half image have overlapping pixels with its registered contour and how many pixels they have in common. From what this graph illustrates, the first contour of the right part (see figure 4.21(c)) is its paired contour because of sharing more overlapping pixels. Figure 4.22 shows that the third largest contour of the left part does not correspond to any contours of the right side. This may also be inferred from figure 4.23 and figure 4.24. The fourth contour and the fifth contour of the left half image correspond to the third contour and the seventh contour of the right half image, respectively. Figure 4.25(a) shows which contour of the left side corresponds to which contour from right half image. The result of starting from the contours of the right half image can be seen in figure 4.25(b). By comparing these two results, the common pairs were considered as the final matched contour pairs (see 4.25(c)). For this example, the corresponding contours found by starting from the left half are the same as the corresponding contours found by starting from the other half. Figure 4.26 shows all final paired contours of the input image. Figure 4.30 illustrates the paired



**Figure 4.20:** (a) The largest contour of the left half image after registration. (b) Overlapping interior pixels between the largest transformed contour (see (a)) and all contours of the reflected right image. (c) The corresponding contour.



**Figure 4.21:** (a) The contour of the left half image after registration. (b) Overlapping interior pixels between the transformed contour (see (a)) and all contours of the reflected right image. (c) The corresponding contour.

contours after applying affine transformation, and figure 4.28(a) shows the local axis of symmetry of each paired contours, before thinning. Finally, figure 4.28(b) shows the result after thinning the symmetry axis.

More results of this proposed method are shown in figures 4.29 to 4.50.

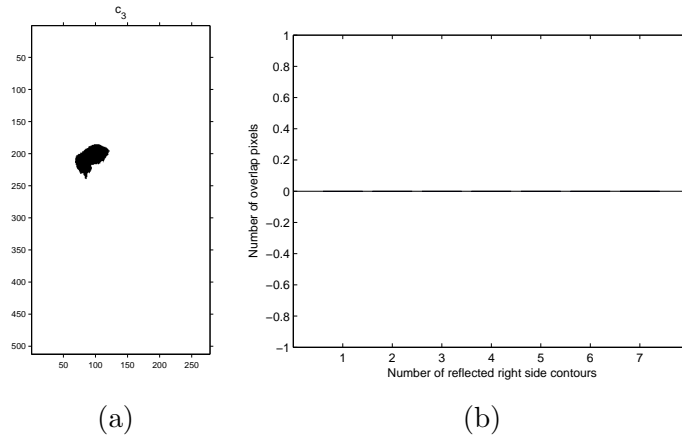


Figure 4.22: (a) The contour of the left half image after registration. (b) Overlapping interior pixels between the transformed contour (see (a)) and all contours of the reflected right image. This contour is not matched to any contour on the right image.

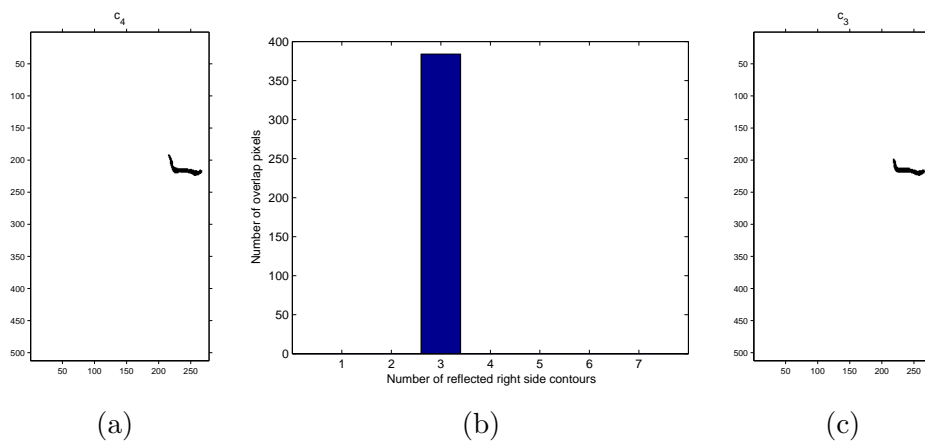


Figure 4.23: (a) The contour of the left half image after registration. (b) Overlapping interior pixels between the transformed contour (see (a)) and all contours of the reflected right image. (c) The corresponding contour.

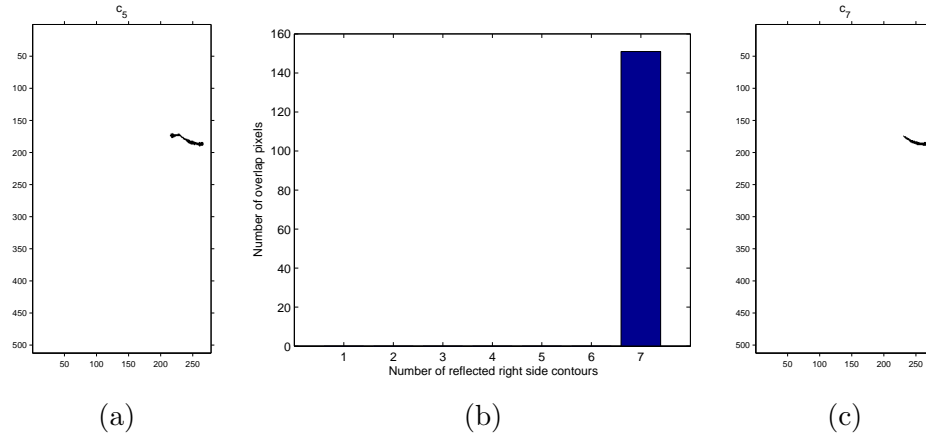


Figure 4.24: (a) The contour of the left half image after registration. (b) Overlapping interior pixels between the transformed contour (see (a)) and all contours of the reflected right image. (c) The corresponding contour.

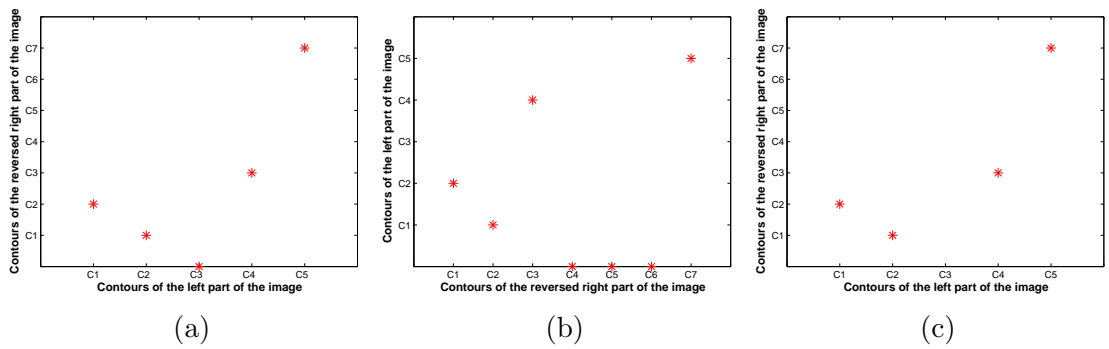


Figure 4.25: (a) Paired contours if the process started from the contours of the left half image. (b) Paired contour if the process started from the contours of the right half image. (c) Final paired contours by comparing the results of (a) and (b).

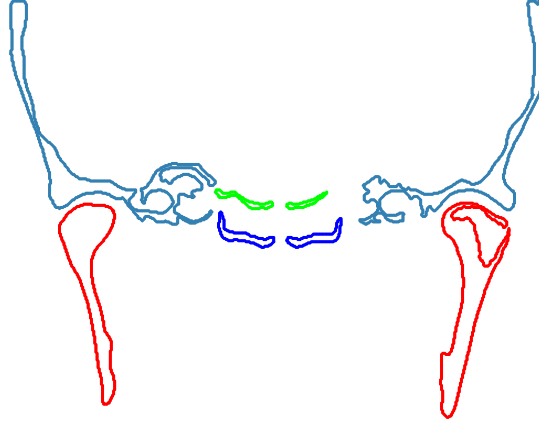


Figure 4.26: Matched paired contours of the input image shown in the same colour

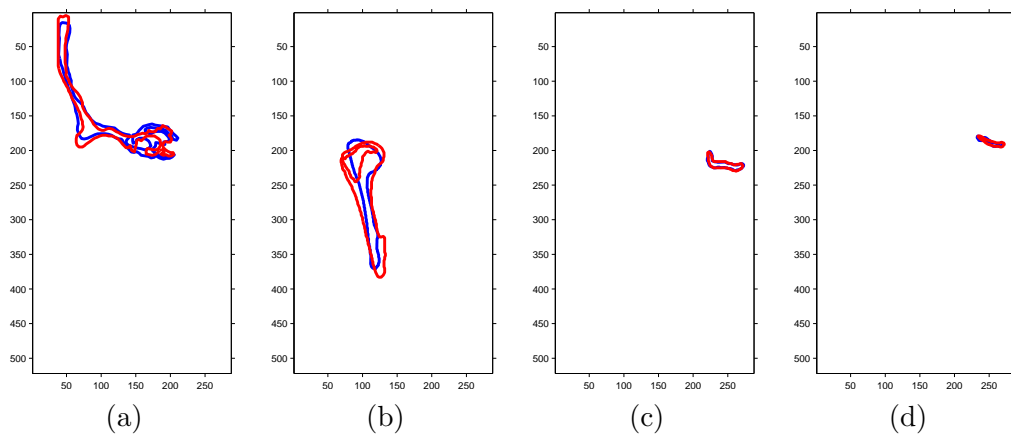


Figure 4.27: Results of affine transformation

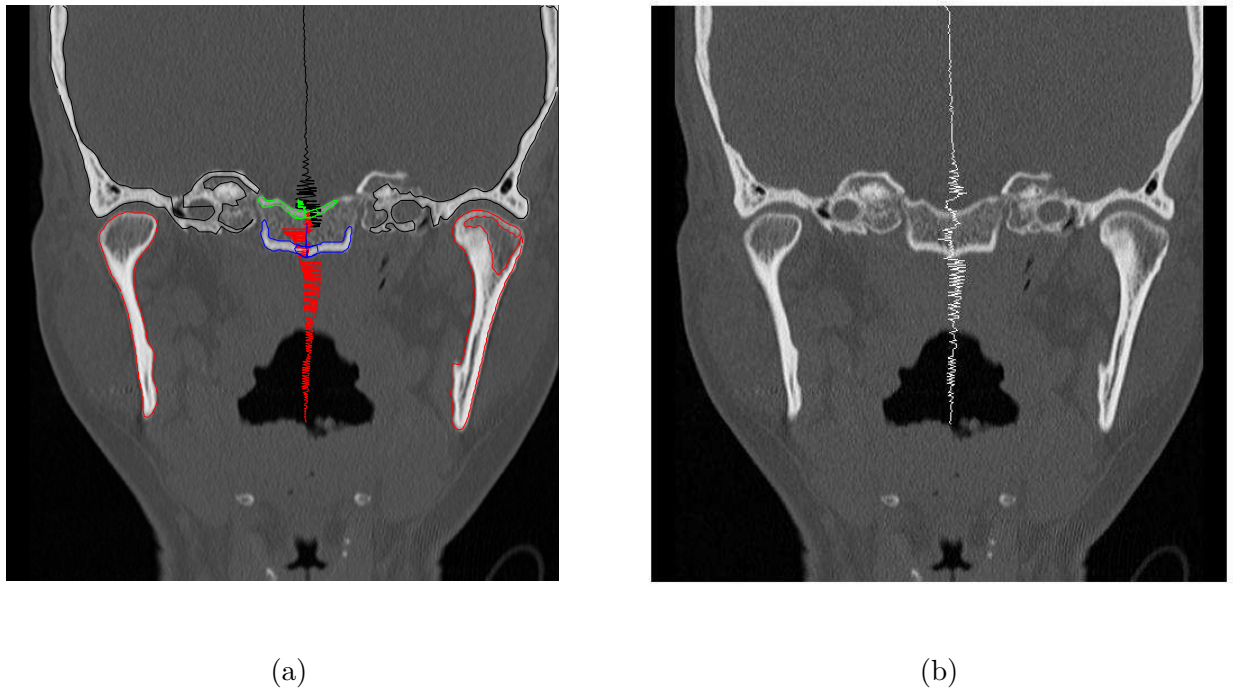


Figure 4.28: (a) Local axis of symmetry of each set of paired contours. (b) Local axis of symmetry after thinning.

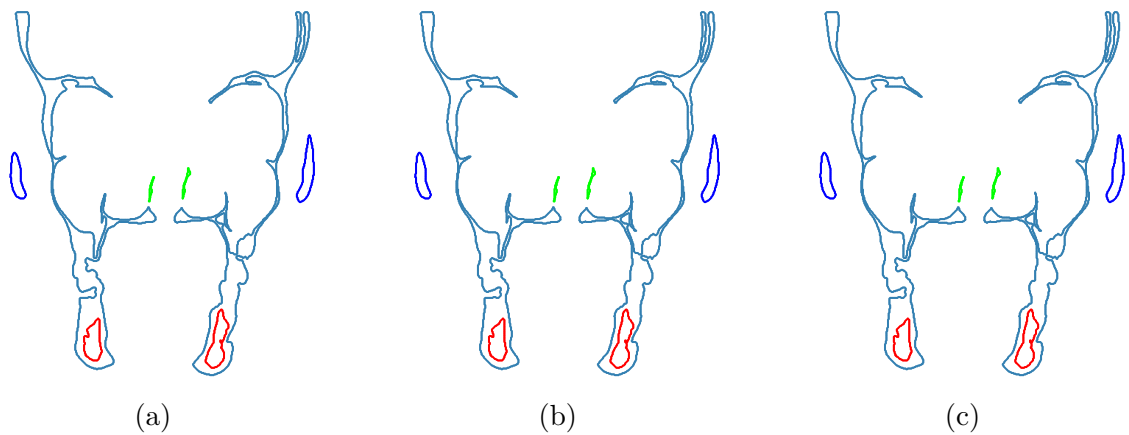


Figure 4.29: (a) Result of paired contours of the image shown in figure 4.7 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

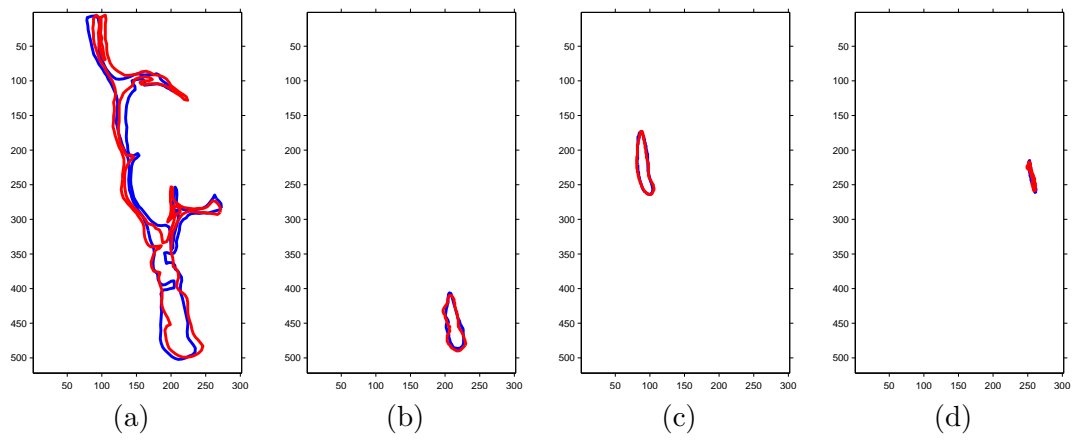
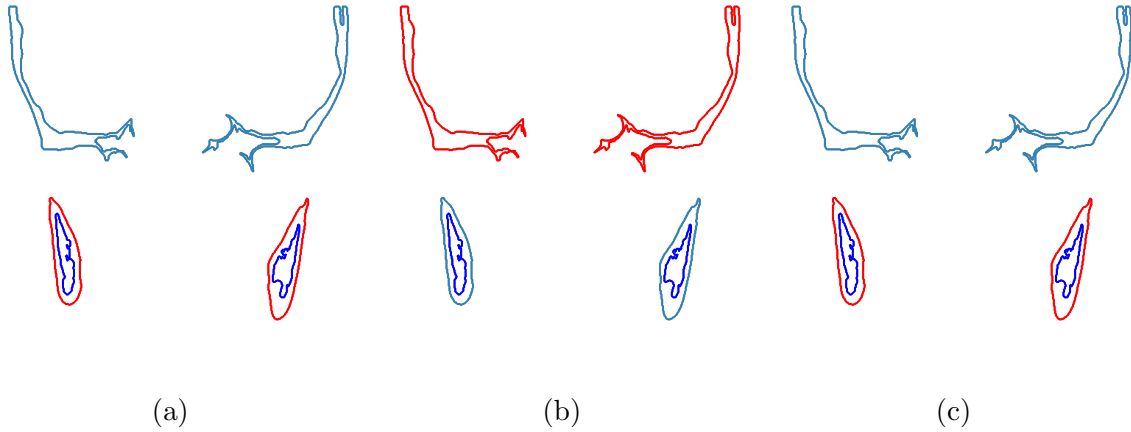


Figure 4.30: Results of affine transformation



Figure 4.31: Local axis of symmetry after thinning



**Figure 4.32:** (a) Result of paired contours of the image shown in figure 4.8 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

#### 4.5.7 Improving the symmetry axis by using a lateral continuity constraint

The previous symmetry method identifies the point of the symmetry axis along each line independent of the point of symmetry axis in the neighbouring lines. This leads to a noisy symmetry axis. In this section, we formulate the problem as one of regularisation, where the symmetry axis is estimated so that its distance from the variable number of candidate points, identified in each line from the multiple contour matched points, is minimised and at the same time the variation of the position of the axis between neighbouring lines is also minimised. For this purpose, we define a cost function. Figure 4.49 shows a possible axis of symmetry plotted against the row index of an image. For each value of  $i$  we may have several candidate points for the axis. Let us call them  $g_1, g_2, \dots, g_{K(i)}$ .  $K(i)$  may be 0, 1, ... for different values of  $i$ . These are the points identified by matching contour points along each line of the image. Let  $x_i$  denote the true position of the axis along row  $i$ . We wish to define all values of  $x_i$  by minimising

$$U \equiv \sum_{i=1}^N \sum_{k=0}^{K(i)} (x_i - g_{ik})^2 + \lambda \sum_{i=1}^{N-1} (x_i - x_{i+1})^2 \quad (4.21)$$

where  $\lambda$  is a parameter that controls the amount of smoothing we impose on the

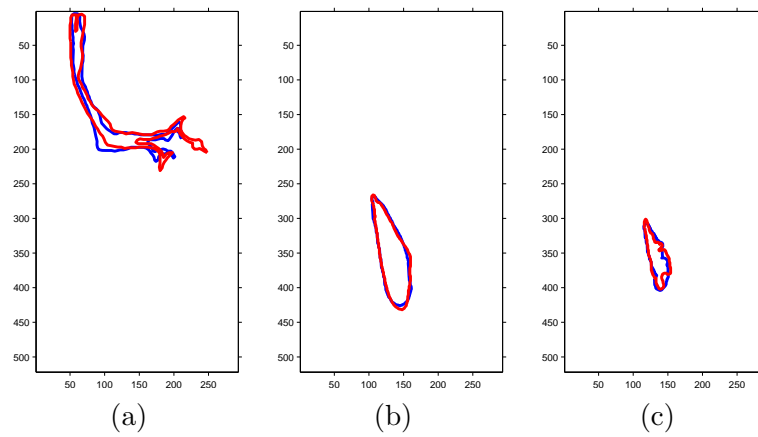


Figure 4.33: Results of affine transformation

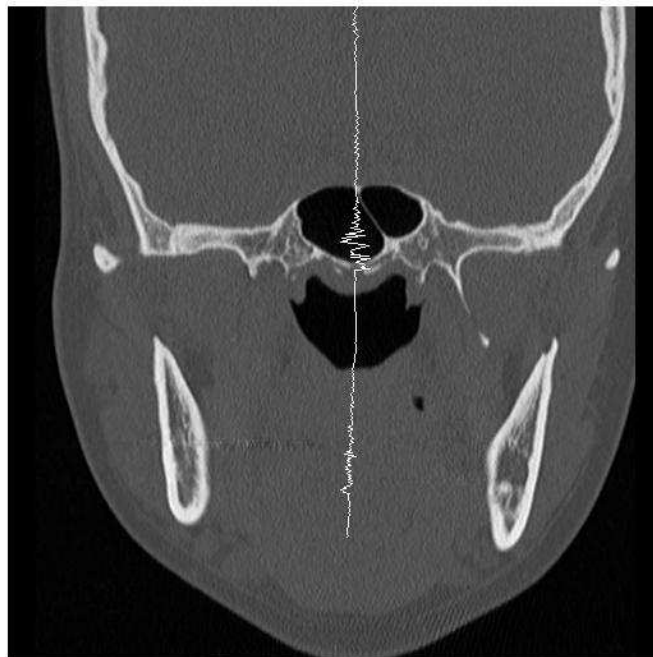


Figure 4.34: Local axis of symmetry after thinning

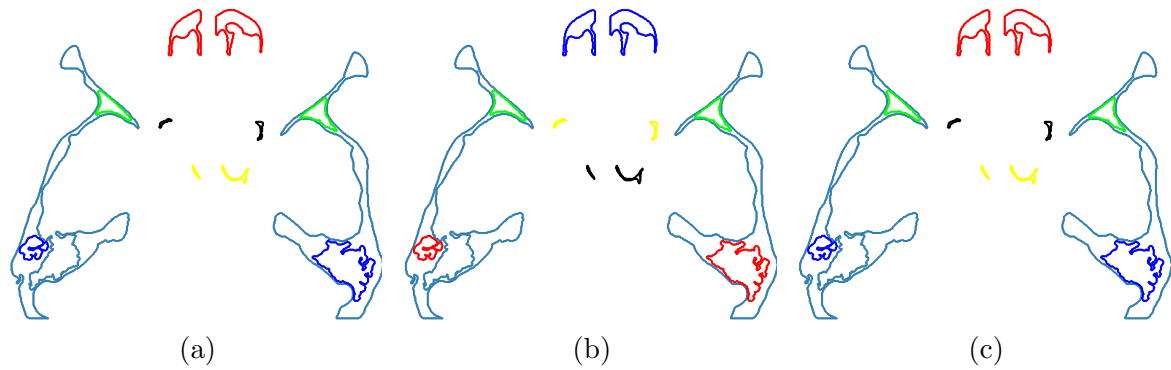


Figure 4.35: (a) Result of paired contours of the image shown in figure 4.10 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

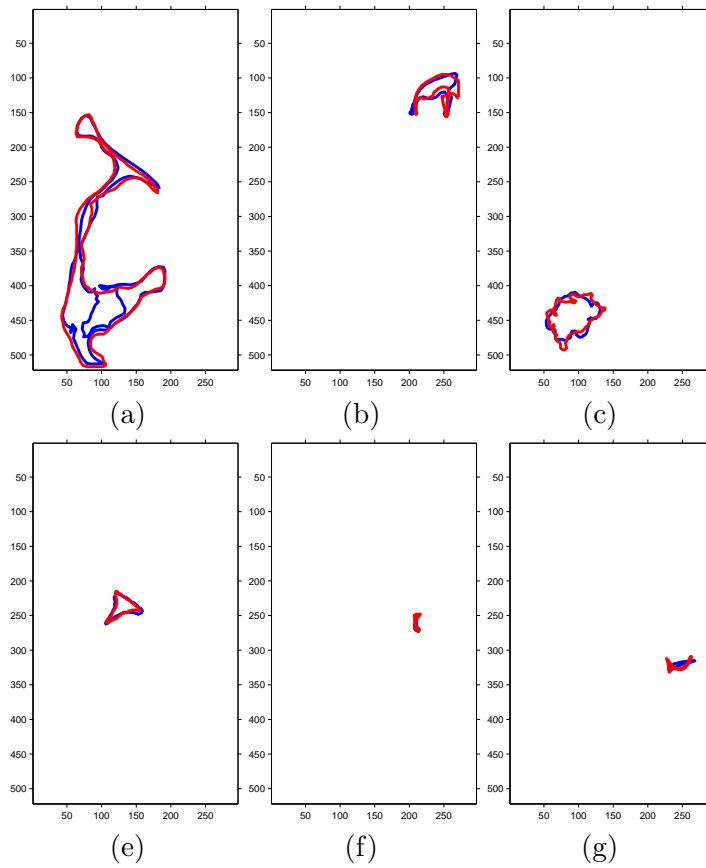


Figure 4.36: Results of affine transformation



Figure 4.37: Local axis of symmetry after thinning

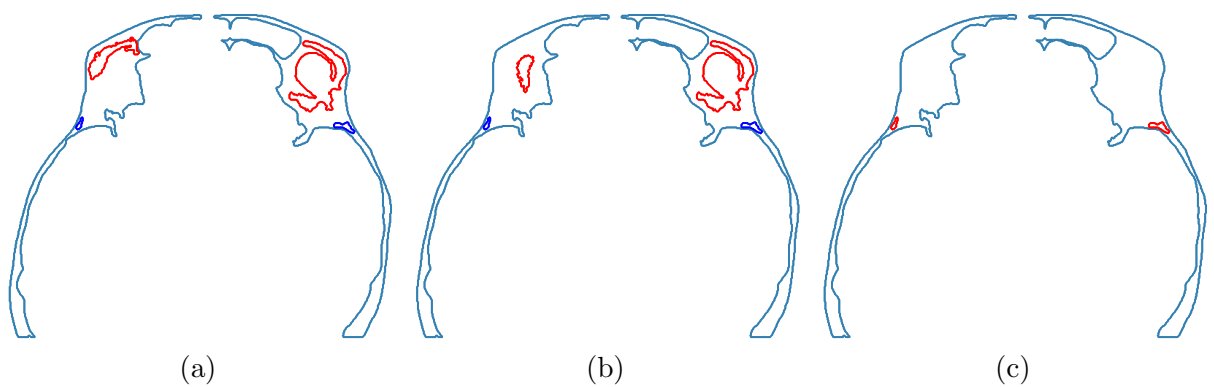


Figure 4.38: (a) Result of paired contours of the image shown in figure 4.11 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

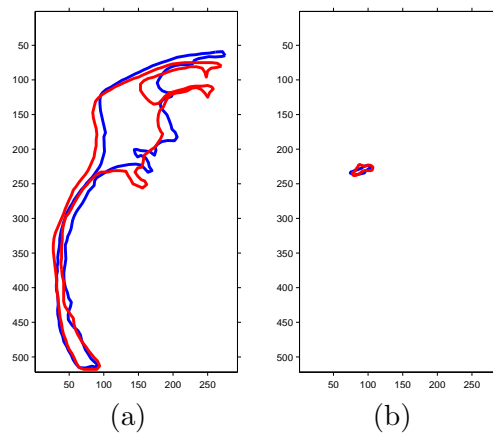


Figure 4.39: Results of affine transformation



Figure 4.40: Local axis of symmetry after thinning

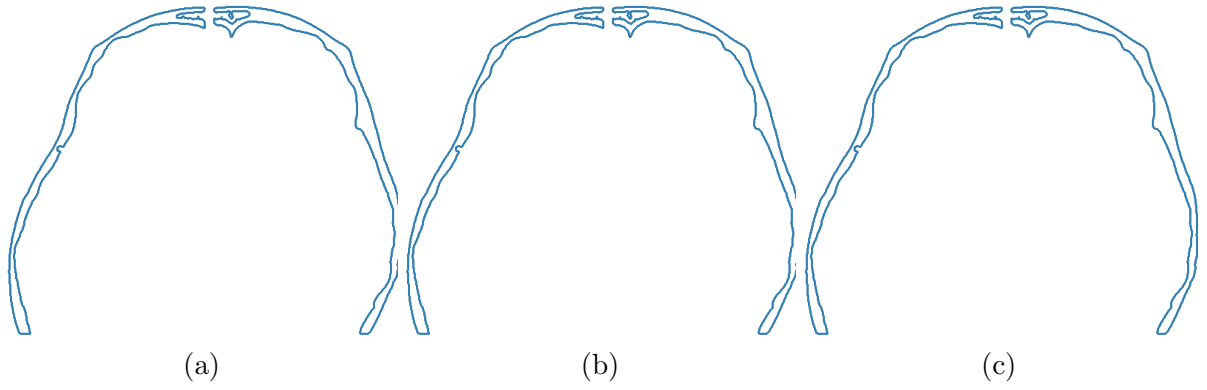


Figure 4.41: (a) Result of paired contours of the image shown in figure 4.12 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

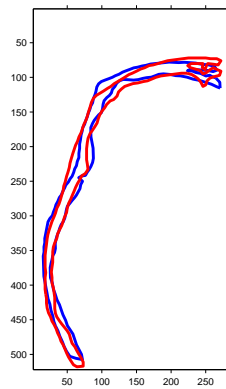


Figure 4.42: Results of affine transformation

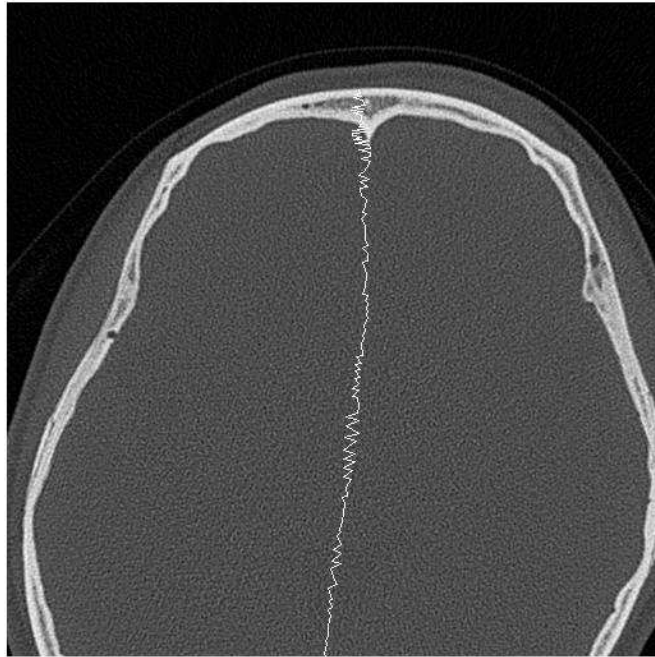


Figure 4.43: Local axis of symmetry after thinning

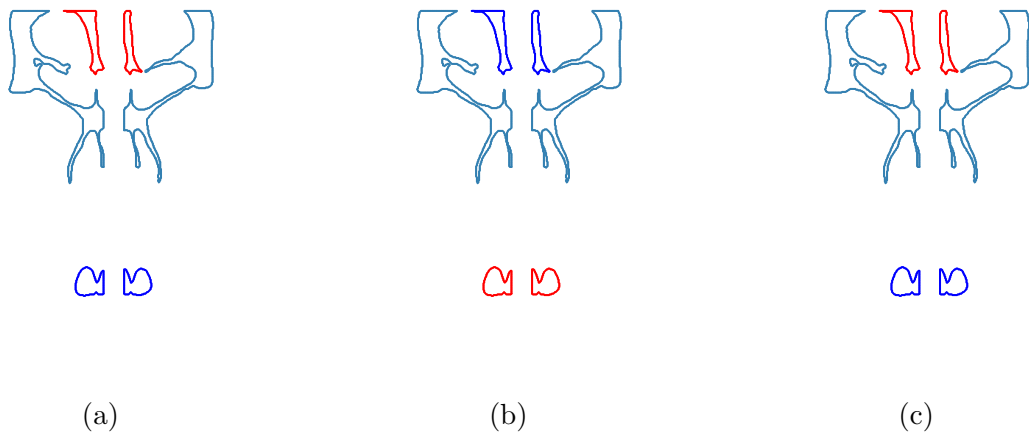


Figure 4.44: (a) Result of paired contours of the image shown in figure 4.13 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

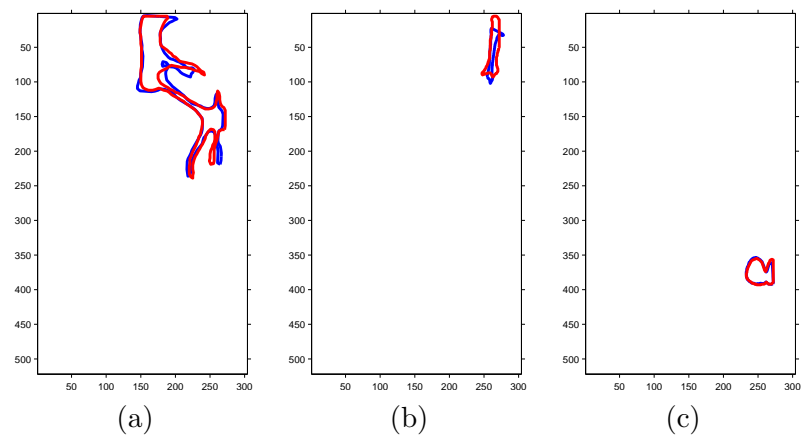


Figure 4.45: Results of affine transformation



Figure 4.46: Local axis of symmetry after thinning

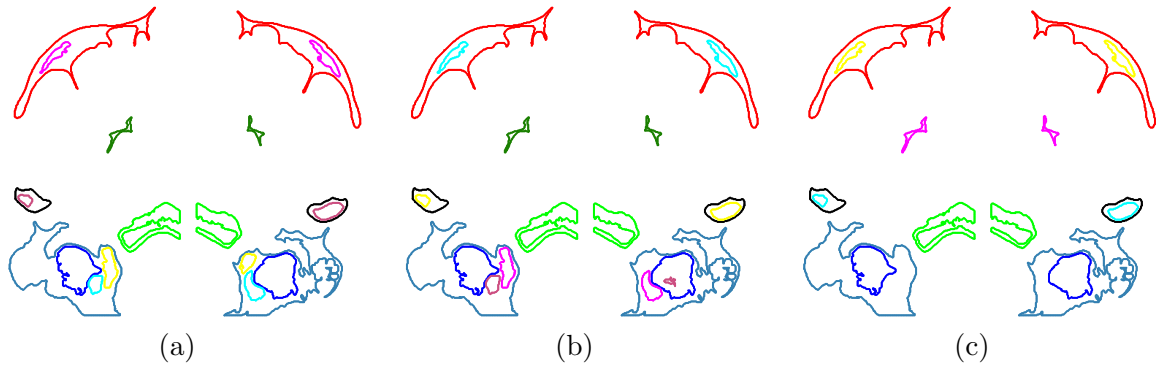


Figure 4.47: (a) Result of paired contours of the image shown in figure 4.14 if matching started from contours of the left half image (paired contours are displayed with the same colour). (b) Result of paired contours if matching started from contours of the reverse right half image (paired contours are displayed with the same colour). (c) Final result.

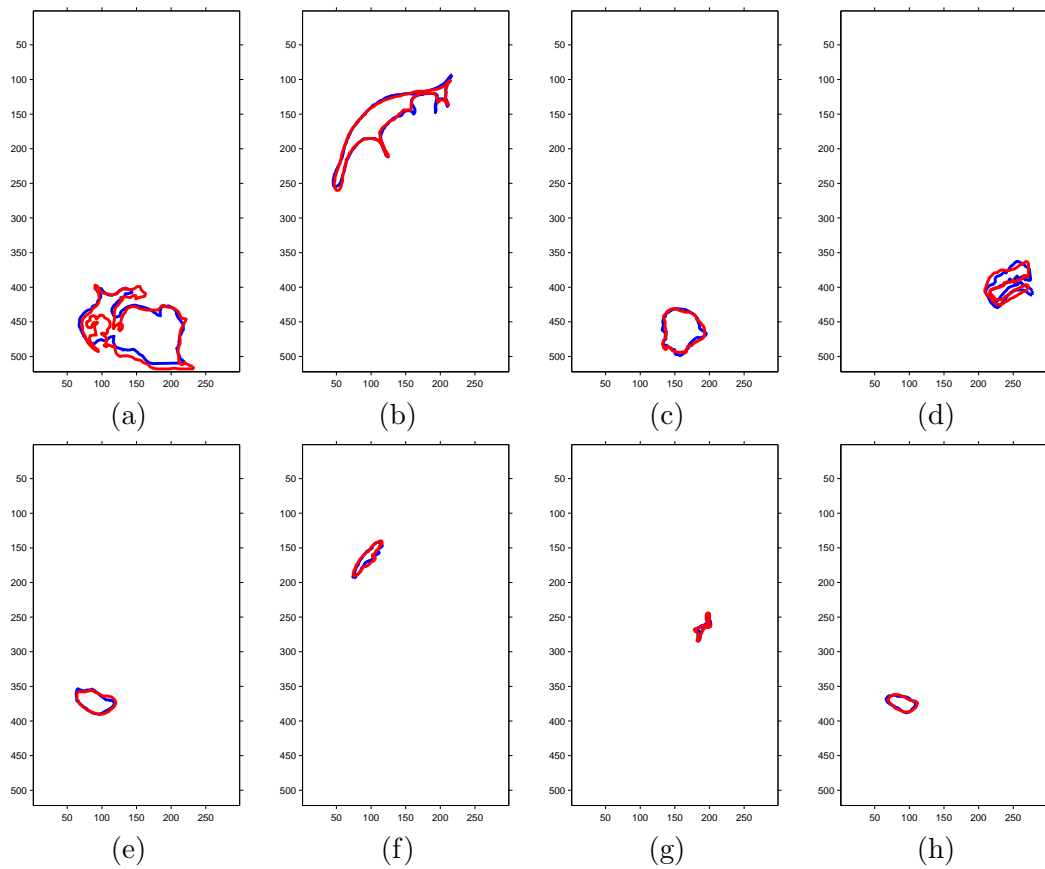


Figure 4.48: Results of affine transformation

axis. This is a quadratic function in the unknowns and the true position  $x_i$  may be estimated by differentiating the quadratic function.

$$U = \sum_{i=1}^N \sum_{k=0}^{K(i)} (x_i^2 + g_{ik}^2 - 2 x_i g_{ik}) + \lambda \sum_{i=1}^{N-1} (x_i^2 + x_{i+1}^2 - 2 x_i x_{i+1}) \quad (4.22)$$

$$U = \sum_{i=1}^N \left[ K(i) x_i^2 + \sum_{k=0}^{K(i)} g_{ik}^2 - 2 x_i \sum_{k=0}^{K(i)} g_{ik} \right] + \lambda \sum_{i=1}^{N-1} x_i^2 + \lambda \sum_{i=1}^{N-1} x_{i+1}^2 - 2\lambda \sum_{i=1}^{N-1} x_i x_{i+1} \quad (4.23)$$

$\sum_{i=1}^N K(i) x_i^2 + \lambda \sum_{i=1}^{N-1} x_i^2$  may be written as  $K(N) x_N^2 + \sum_{i=1}^{N-1} ((K(i) + \lambda) x_i^2)$ , so

$$U = K(N) x_N^2 + \sum_{i=1}^{N-1} ((K(i) + \lambda) x_i^2) + \sum_{i=1}^N \sum_{k=0}^{K(i)} g_{ik}^2 - \sum_{i=1}^N 2 x_i \sum_{k=0}^{K(i)} g_{ik} + \lambda \sum_{i=1}^{N-1} x_{i+1}^2 - 2\lambda \sum_{i=1}^{N-1} x_i x_{i+1} \quad (4.24)$$

Change variable of summation in the 5<sup>th</sup> term:  $\tilde{i} \equiv i + 1$ . Then  $\sum_{i=1}^{N-1} x_{i+1}^2 = \sum_{\tilde{i}=2}^N x_{\tilde{i}}^2 = \sum_{i=2}^N x_i^2$ . By rewriting  $\lambda \sum_{i=1}^{N-1} x_{i+1}^2$  as  $\lambda \sum_{i=2}^N x_i^2$  which is equal to  $\lambda x_N^2 + \lambda \sum_{i=2}^{N-1} x_i^2$ , and also by rewriting  $\sum_{i=1}^{N-1} ((K(i) + \lambda) x_i^2)$  as  $(K(1) + \lambda) x_1^2 + \sum_{i=2}^{N-1} ((K(i) + \lambda) x_i^2)$ , equation (4.24) becomes

$$U = K(N) x_N^2 + (K(1) + \lambda) x_1^2 + \lambda x_N^2 + \sum_{i=2}^{N-1} ((K(i) + 2\lambda) x_i^2) + \sum_{i=1}^N \sum_{k=0}^{K(i)} g_{ik}^2 - 2 \sum_{i=1}^N x_i \sum_{k=0}^{K(i)} g_{ik} - 2\lambda \sum_{i=1}^{N-1} x_i x_{i+1} \quad (4.25)$$

If we define  $A \equiv \sum_{i=1}^N \sum_{k=0}^{K(i)} g_{ik}^2$  and  $B(i) \equiv \sum_{k=0}^{K(i)} g_{ik}$ , then

$$U = (K(N) + \lambda) x_N^2 + (K(1) + \lambda) x_1^2 + \sum_{i=2}^{N-1} ((K(i) + 2\lambda) x_i^2) + A - 2 \sum_{i=1}^N x_i B(i) - 2\lambda \sum_{i=1}^{N-1} x_i x_{i+1} \quad (4.26)$$

For  $U$  to be minimum, its derivative with respect to all independent variables must be

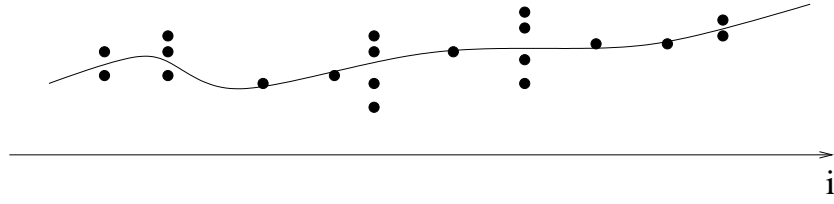


Figure 4.49: Locations of multiple symmetry points

zero:

$$\frac{\partial U}{\partial x_N} = 2(K(N) + \lambda)x_N - 2B(N) - 2\lambda x_{N-1} = 0 \quad (4.27)$$

$$\frac{\partial U}{\partial x_1} = 2(K(1) + \lambda)x_1 - 2B(1) - 2\lambda x_2 = 0 \quad (4.28)$$

If  $i \neq 1$  and  $i \neq N$

$$\frac{\partial U}{\partial x_i} = 2(K(i) + 2\lambda)x_i - 2B(i) - 2\lambda x_{i+1} - 2\lambda x_{i-1} = 0 \quad (4.29)$$

The above equations represent a set of linear equations. Therefore, these equations may be written in matrix form as

$$SX = F \Rightarrow X = S^{-1}F \quad (4.30)$$

where  $S$ ,  $X$  and  $F$  are defined as

$$S = \begin{bmatrix} K(1) + \lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & K(2) + 2\lambda & -\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -\lambda & K(N-1) + 2\lambda & -\lambda \\ 0 & \cdots & 0 & -\lambda & K(N) + \lambda \end{bmatrix}_{N \times N} \quad (4.31)$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}_{N \times 1} \quad (4.32)$$

$$F = \begin{bmatrix} B(1) \\ B(2) \\ \vdots \\ B(N) \end{bmatrix}_{N \times 1} \quad (4.33)$$

This method will be used in the finally adopted scheme we shall evaluate in the next chapter.

#### 4.5.8 Straightening the symmetry axis and identifying the abnormalities

As noted previously, reflective symmetry is one of the main characteristics of the human body that may be used as a cue for detecting and identifying abnormalities or deformities. Therefore, to indicate any suspicious regions in the medical image, it is necessary to compare the two sides of the image divided by the extracted axis of symmetry. Moreover, to facilitate comparison of the left side with the reflected right side of the image, the local axis of symmetry may be made straight by alignment of the image pixels. The alignment can be done by shifting the pixels of each line so that the points of reflective symmetry of all image lines are aligned. Then, to identify a deformity area, the absolute difference of the left side and the reflected right side of the imaged organ is computed. Note that small misalignments along the boundaries of the various tissues may appear as differences between the two halves. To identify the significant differences, grey scale opening is used to remove thin lines from the difference image, and enhance the significant differences. Figure 4.51 shows the example of an image with facial trauma where the trauma region and the swelling of one side of the head are identified.

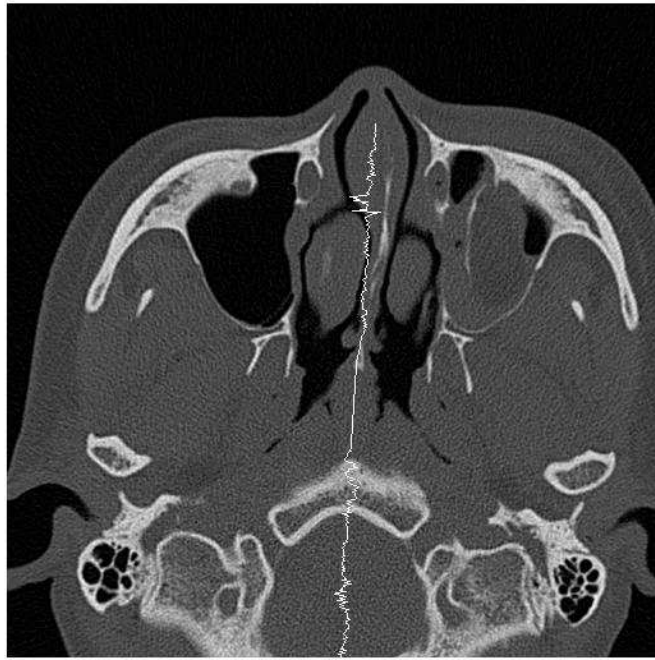


Figure 4.50: Local axis of symmetry after thinning



(a)



(b)

Figure 4.51: (a) Absolute difference of two halves of the image of a head. The result has been scaled so white corresponds to zero difference. (b) After applying grey scale opening with a  $15 \times 15$  structuring element.

## 4.6 Conclusions from this chapter

This chapter described a new method for determining the symmetry axis of an imaged organ starting from a gross, straight line estimate and refining it by using sub-part matching. First, this chapter gave some literature review of previous work on symmetry detection. Then, it described the methodology in detail. The proposed method combines registration and extraction of bone contours to generate the desired symmetry axis. The method consists of several stages: first extracting the bone contours of the images by using an active contour method, then finding grossly an axis that allows the division of the image into two parts, loosely called “half” images, but with the understanding that they are not exactly the two halves of the image but rather the two halves of the depicted object. After that, finding a mapping that aligns the first half of the image with the second half and finally, finding the local axis of symmetry from corresponding contours.

## Chapter 5

# Evaluation

### 5.1 Introduction

**A**S mentioned in the previous chapter, our algorithm consists of two parts: 1) extracting the axis of symmetry in CT images, and 2) identifying deformity areas in CT images. A total of 6 subject images (more than 1000 individual images) with different types of injury were selected for a validation study of our algorithm.

### 5.2 Results

First, the algorithm was applied on a CT scan facial series (405 images) of a patient with a left orbital blowout fracture. Figure 5.1(a) shows some of the input images and figure 5.1(b) shows the symmetry axis extracted by applying our method. The results of obtaining straight axes of symmetries after the pixel alignment process can be seen in figure 5.2(a). Figure 5.2(b) shows the results of applying the method for identifying the deformity. The white marks show the difference between the two halves of the images which in this case correctly identify the orbital blowout fracture. To validate the extraction of the axes of symmetry method, we computed the correlation coefficient of the two halves of the input images before correction (original) and after the final alignment. The result of the comparison is shown in figure 5.3.

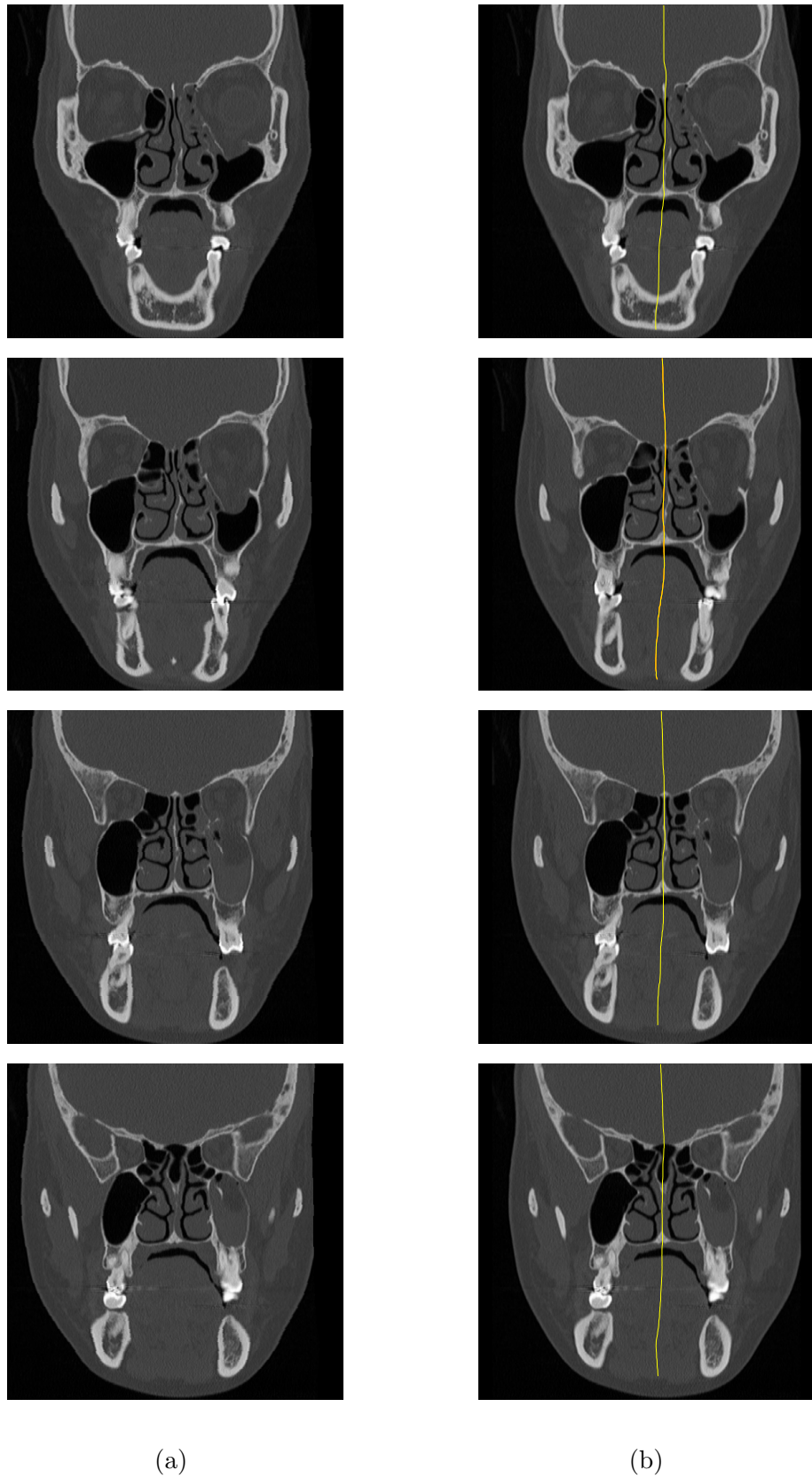


Figure 5.1: Orbital blowout fracture case (a) Input images. (b) Local axis of symmetry.

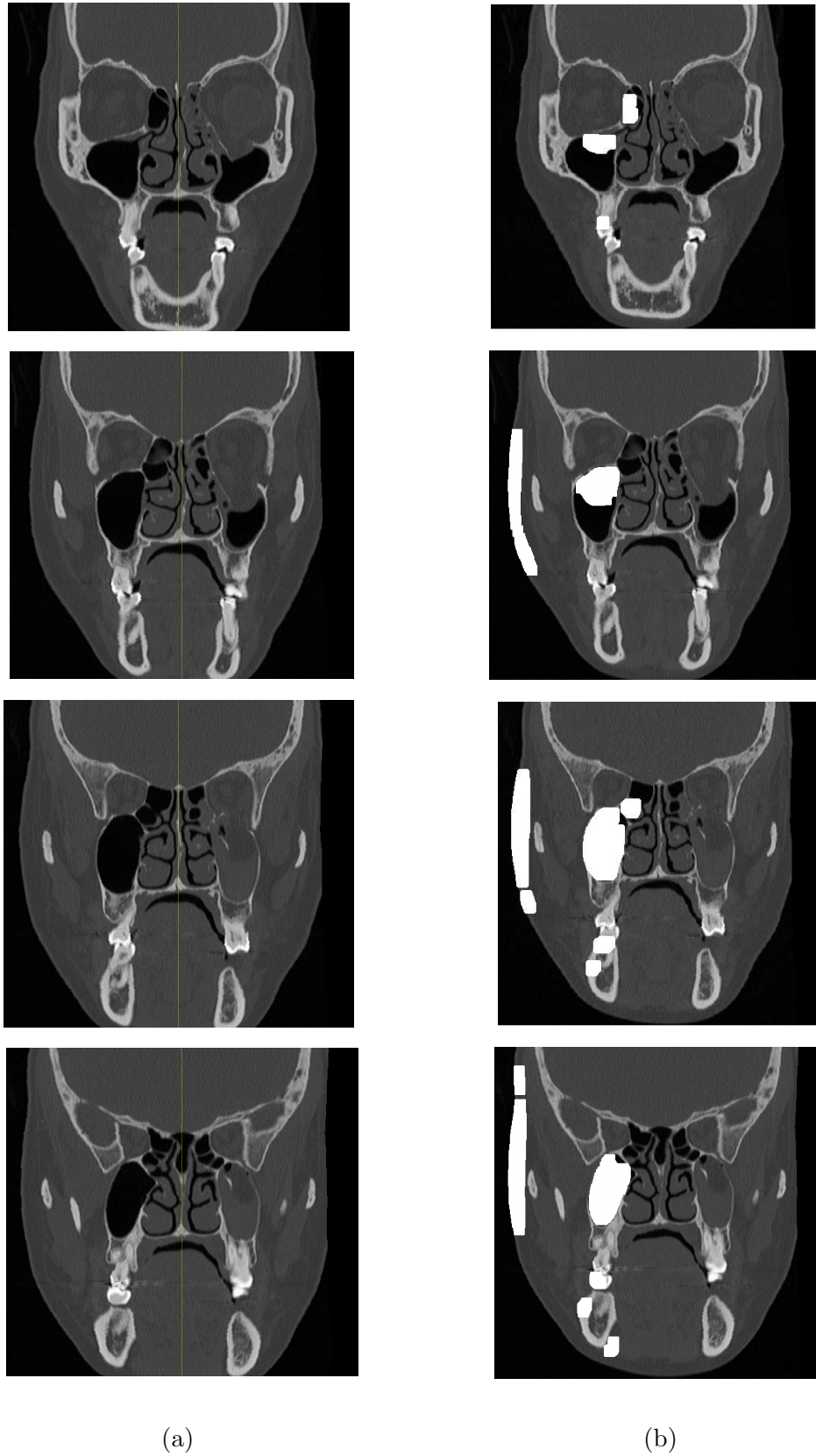
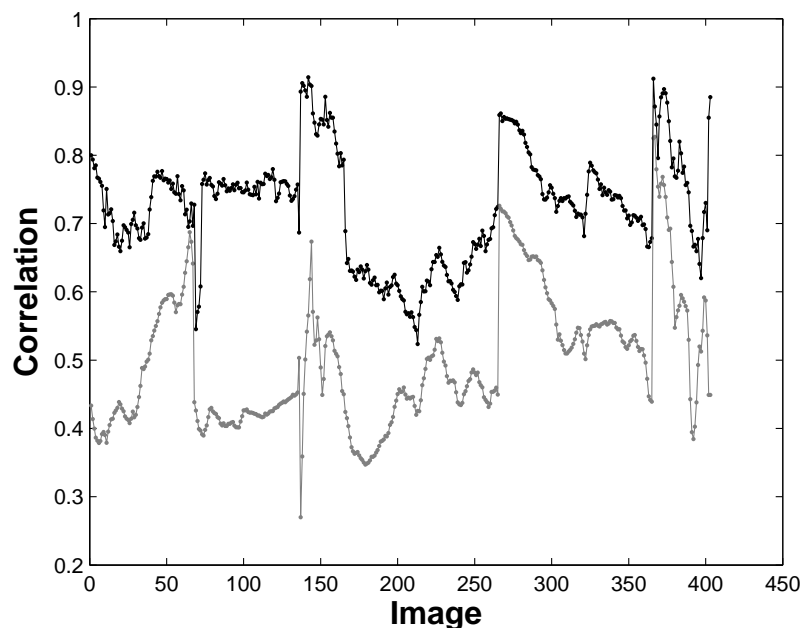


Figure 5.2: Orbital blowout fracture case (a) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.

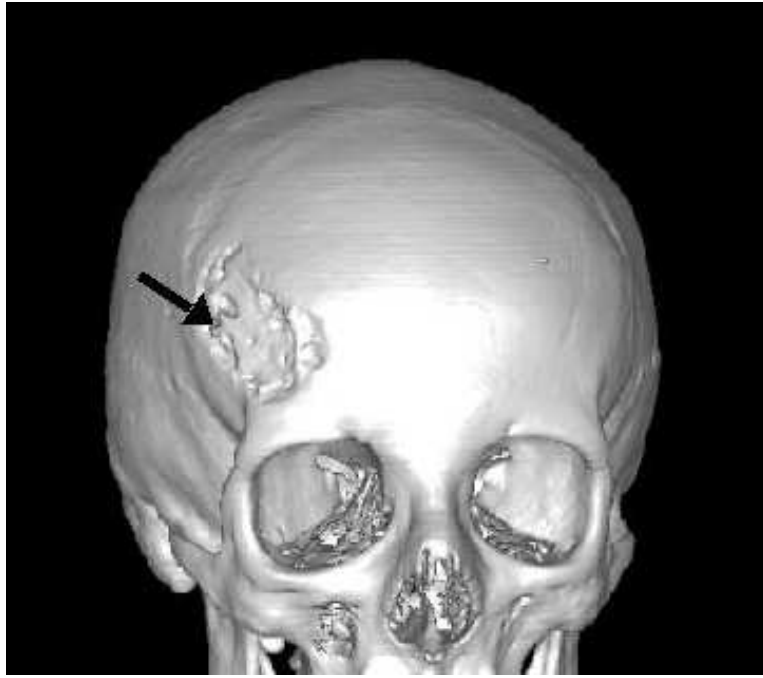


**Figure 5.3:** Correlation of two image halves before correction (gray) and after correction (black) for 405 test images of the first patient.

The second subject was a female patient with frontotemporal osteoradionecrosis. The CT scan was taken for planning titanium cranioplasty. The 3D image of the skull of the patient is shown in figure 5.4. The black arrow in the 3D image indicates the defected area. The results of our algorithm are shown in figure 5.5. Figure 5.5(a) shows two CT images of the skull of a female patient. In 5.5(b), the results of applying our proposed algorithm for detecting axes of symmetries are shown. Figure 5.5(c) shows the straight axis of symmetries and finally, in 5.5(d), the bone defected area is detected and shown in white. Also, figure 5.6 shows the validation of the extraction of the axis of symmetry method by computing the correlation coefficient of the two halves of the input images before correction (original) and after the final alignment.

Next, the algorithm was applied on a CT scan facial series of a patient with multiple fractures on his skull. Figures 5.7 and 5.8 show the results for this patient. As it can be seen, the multiple bone fractures and soft tissue deformities are identified by our method.

The fourth case was a patient with a Nasal bone fracture. Figure 5.9 shows the



**Figure 5.4:** The black arrow indicates the defected area

results for this case. In addition, the results of the validation process can be seen in figure 5.10.

Fronto orbital neurofibromatosis case was the fifth case that our algorithm was applied to. The results are shown in figures 5.11 and 5.12.

Finally, our algorithm was applied to a normal case. The results are shown in figures 5.13 and 5.14.

### 5.3 Discussion and Conclusions

We presented some generic methodology for defining a generalised symmetry axis in medical images. After detecting the symmetry axis, the difference between the two halves of the imaged organ were quantified. The only assumption we made was that the axis is roughly vertical. This, however, is not particularly restrictive as frontal imaging produces such images and in any case images may be rotated prior to using our algorithm to comply with this assumption. The application we have in mind is plastic surgery for correcting

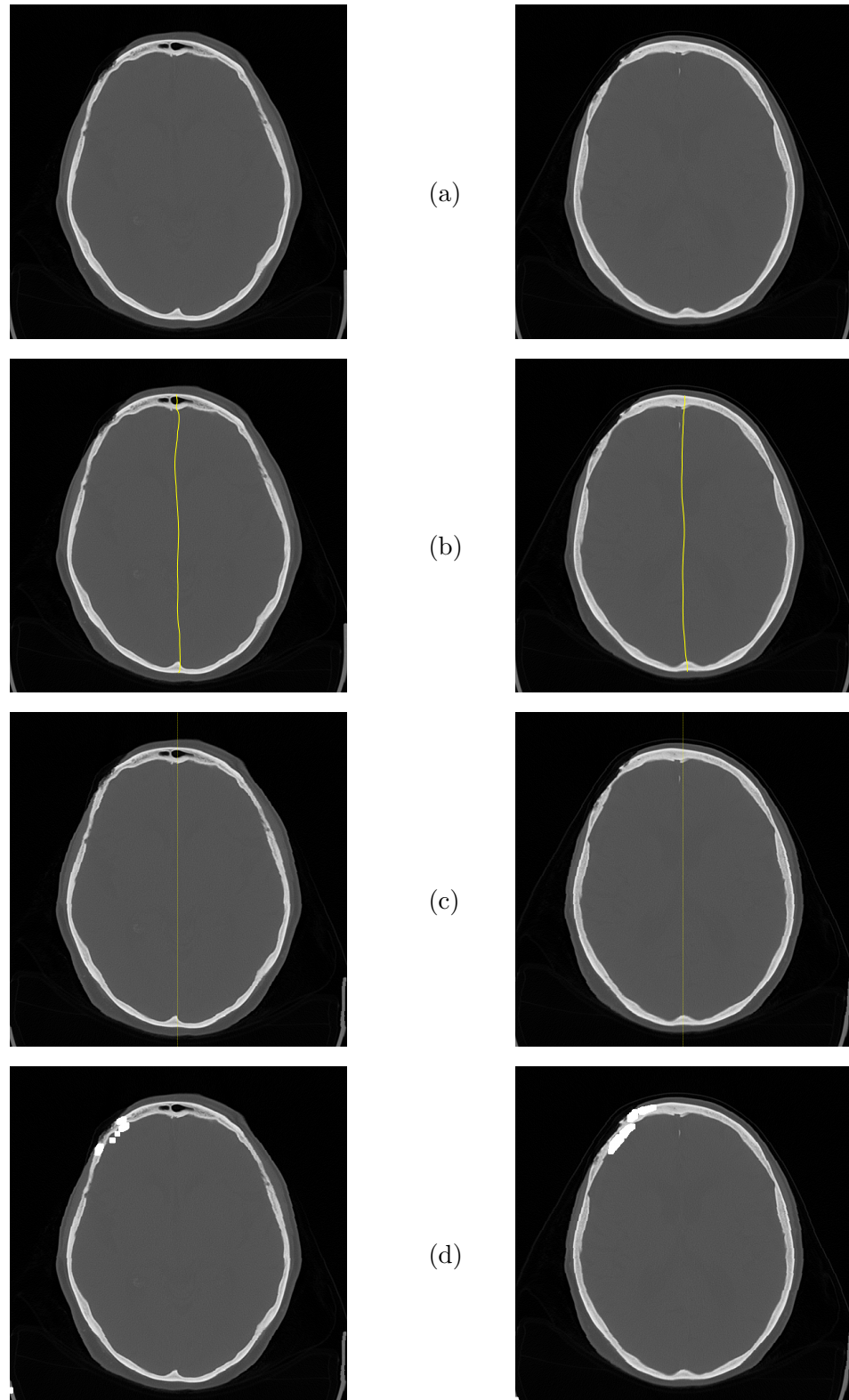


Figure 5.5: Frontotemporal bone defect case (a) Input images. (b) Local axis of symmetry. (c) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.

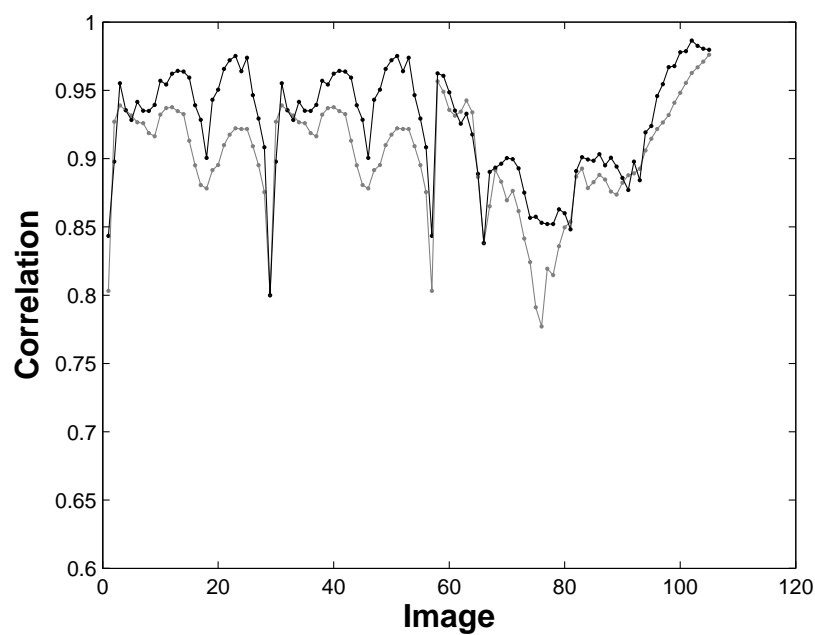


Figure 5.6: Correlation of two image halves before correction (gray) and after correction (black) for 107 test images of the second patient.

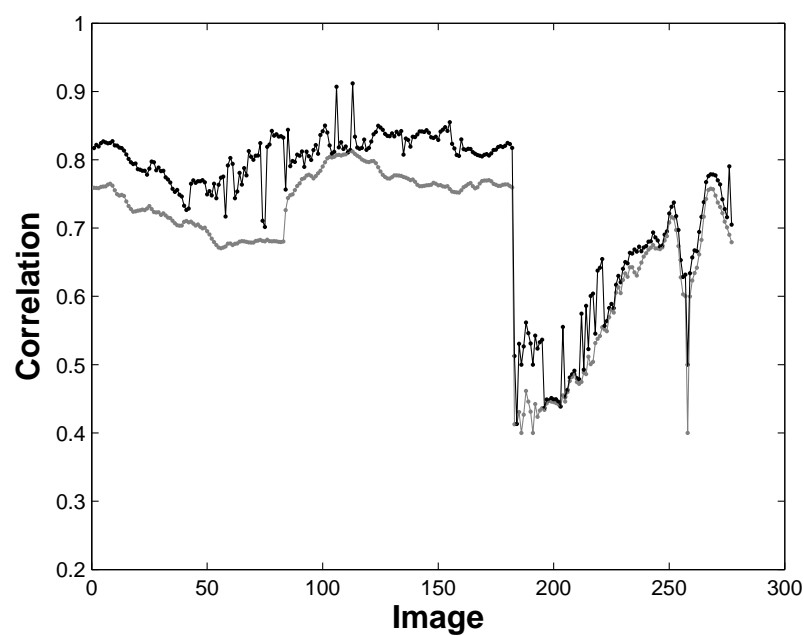
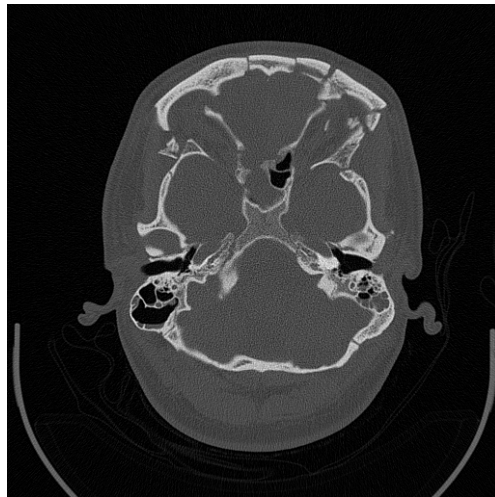
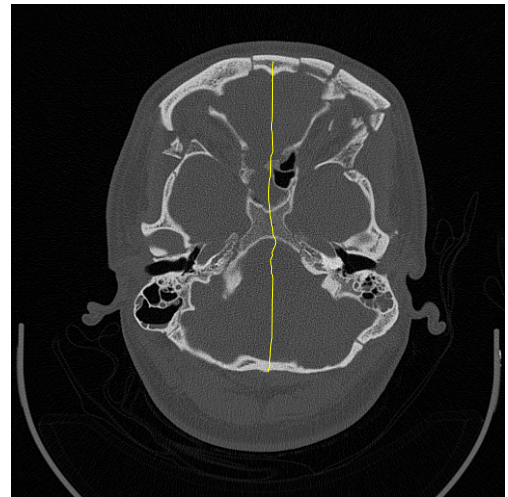


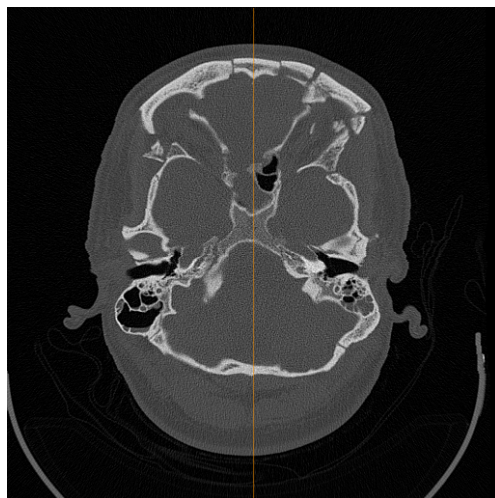
Figure 5.7: Correlation of two image halves before correction (gray) and after correction (black) for the third patient CT images



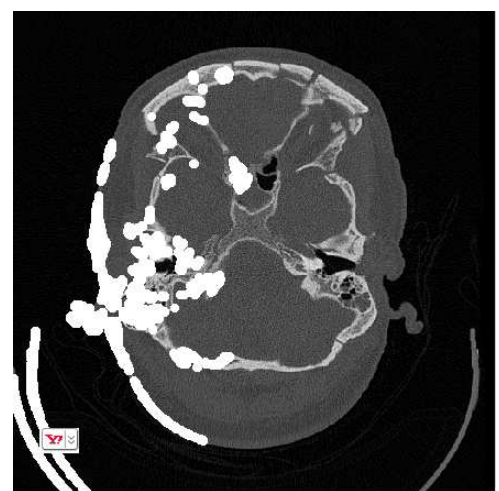
(a)



(b)

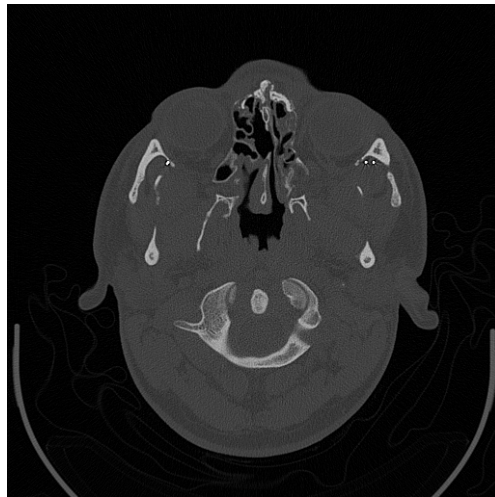


(c)

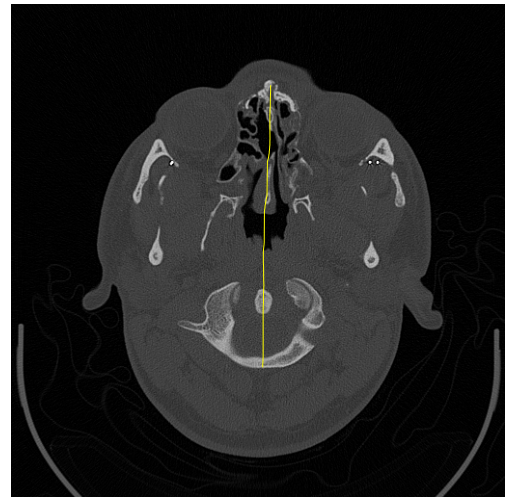


(d)

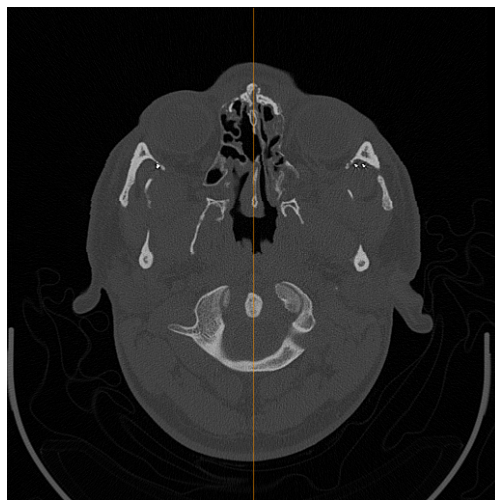
Figure 5.8: Multiple fractures case (a) Input image. (b) Local axis of symmetry. (c) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.



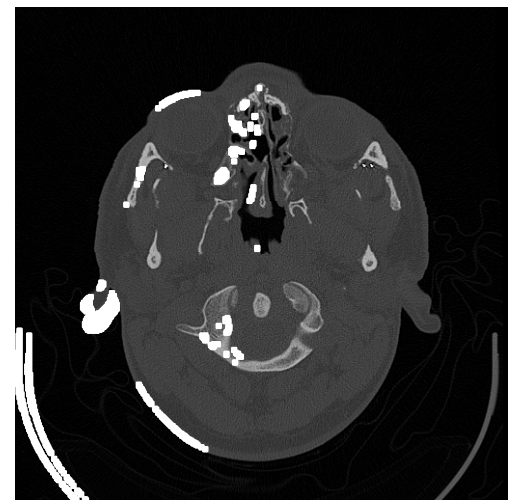
(a)



(b)



(c)



(d)

Figure 5.9: Nasal bone fracture case (a)Input image. (b) Local axis of symmetry. (c) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.

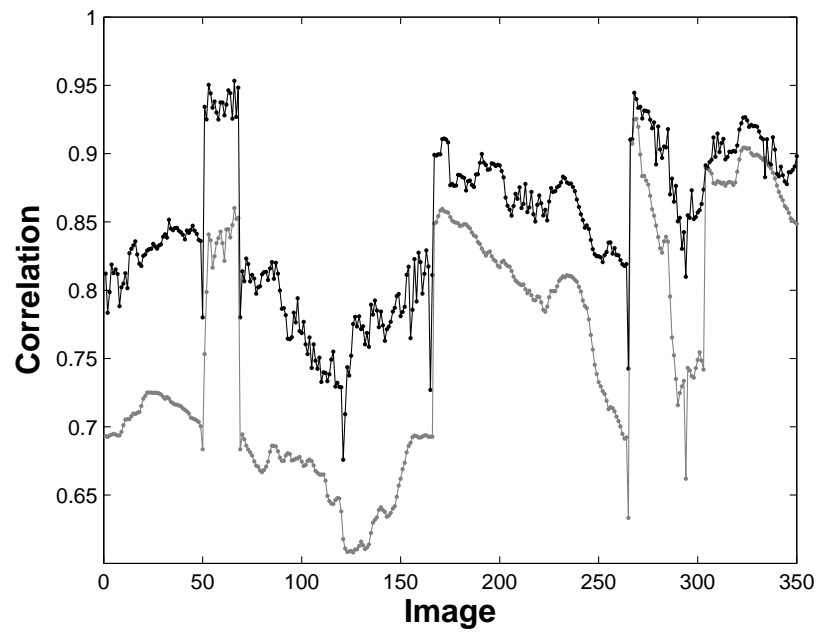


Figure 5.10: Correlation of two image halves before correction (gray) and after correction (black) for CT images of the fourth patient.

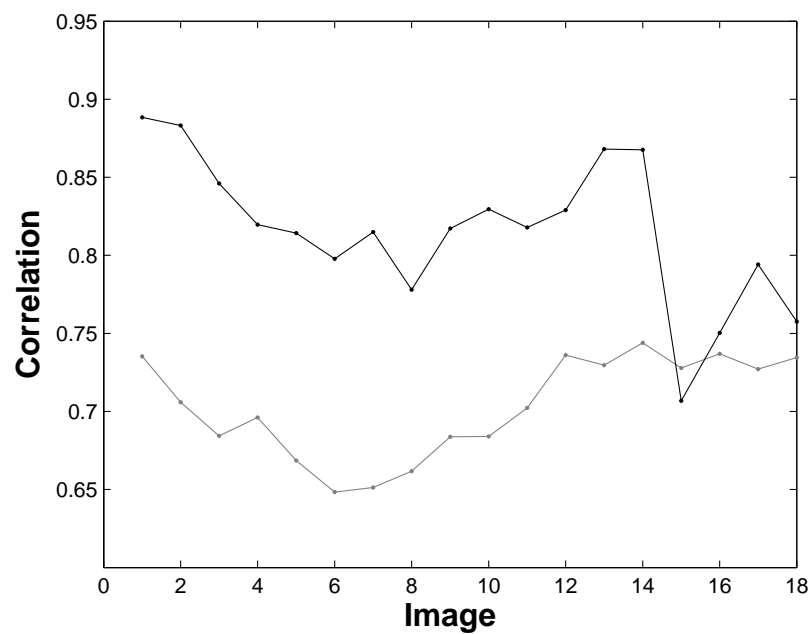
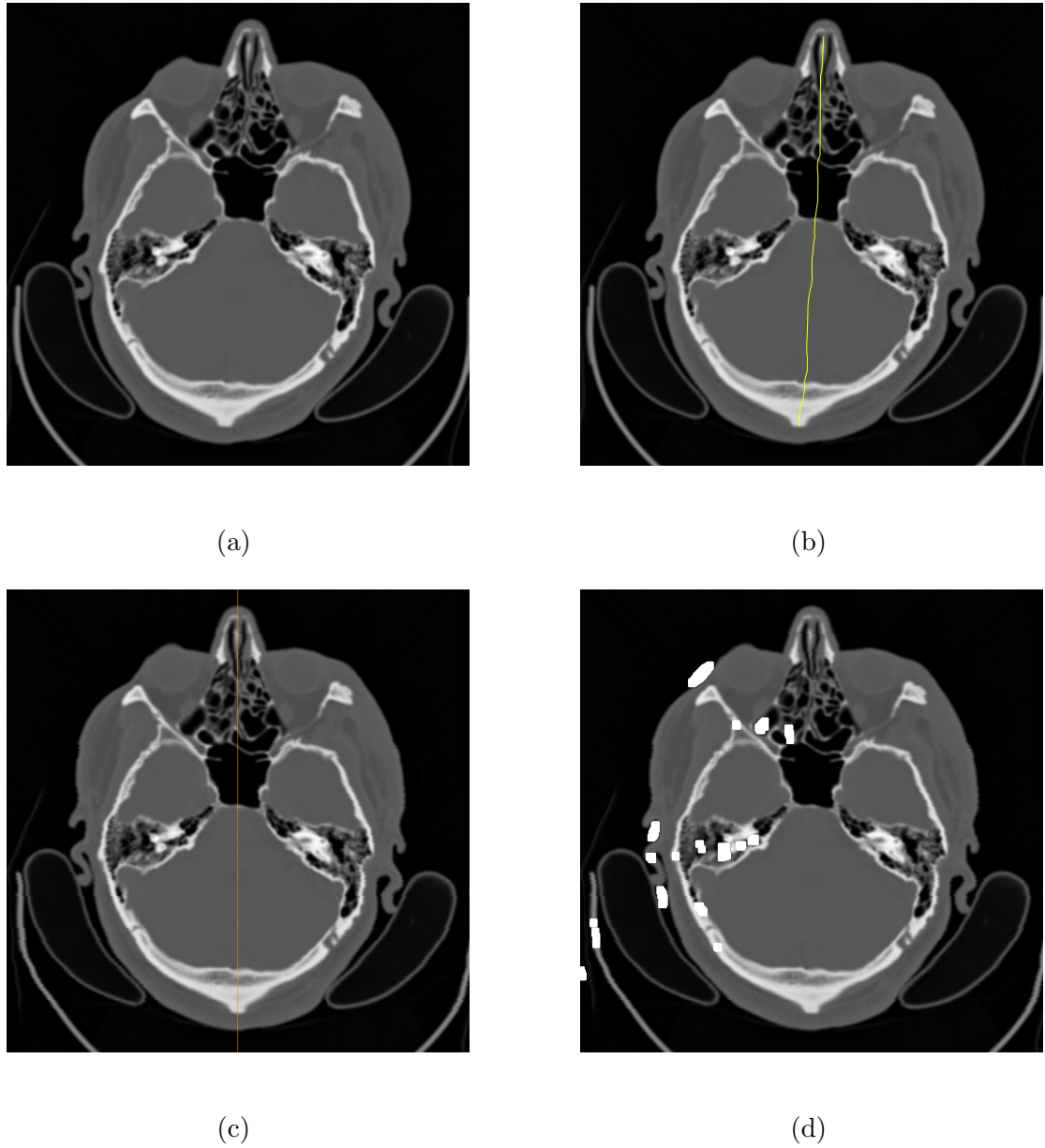
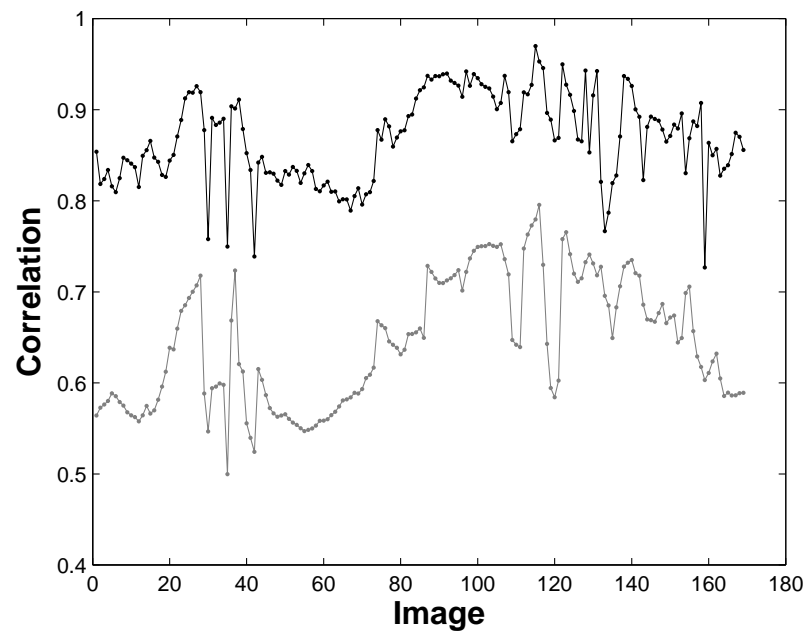


Figure 5.11: Correlation of two image halves before correction (gray) and after correction (black) for CT images of the fifth patient.

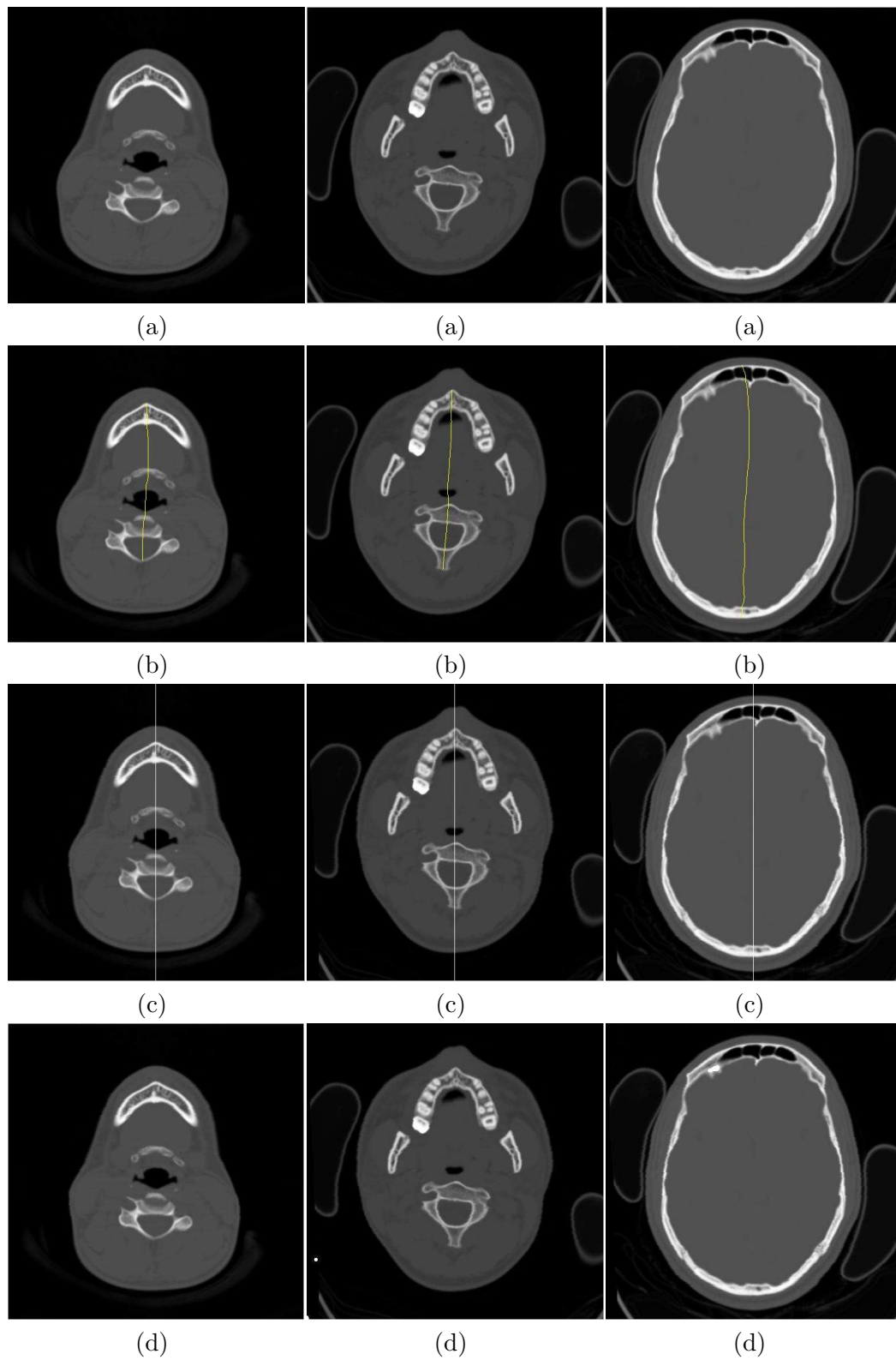


**Figure 5.12:** Fronto orbital neurofibromatosis case (a) Input images. (b) Local axis of symmetry. (c) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.



**Figure 5.13:** Correlation of two image halves before correction (gray) and after correction (black) for CT images of the normal person.

bone deformations. However, the algorithm is also useful for finding defected soft tissues. The method was applied to 1000 images and the detected deformation areas agreed with the doctor's diagnosis, in each case.



**Figure 5.14:** Normal case (a) Input images. (b) Local axis of symmetry. (c) Obtained straight axis of symmetry. (d) White marks indicate suspicious areas.

## Chapter 6

# Conclusions and Future Work

**T**HE purpose of this thesis was to develop some image processing tools to help plastic surgery planning. It was focused on two different fields, one was to identify the cephalometric landmarks on X-ray images and the other was to identify asymmetry in CT images of the human body, relying on the local reflective axis of symmetry.

### 6.1 Cephalometric Landmark Identification

Cephalometry is based on a set of agreed upon feature points (cephalometric landmarks or bony landmarks) from standardized lateral skull radiographs, or cephalograms. Despite all the efforts to automate the process of locating the landmarks, manual locating is still used. The automatic methods are not currently used in clinical routine because of their high failure rate. In the first part of this thesis, we developed methodology to locate cephalometric landmarks on X-ray images, in an automatic way. We proposed an algorithm based on the probabilistic relaxation method. The results show that the proposed algorithm for the identification the cephalometric landmarks is quite successful. Especially, it can be a useful method if its results are considered as an initial identification of the cephalometric landmarks as a time saving first step for the manual identification of the landmarks that happens routinely nowadays. In the future, the algorithm may be

improved if the characteristics of each specific landmark are also considered separately. For instance, by placing a window around each identified landmark and trying to adjust the point based on the surrounding grey level values. Also, different edge detector methods may be used for different landmarks based on whether they are located on strong, continuous edges or not. However, the greatest improvement will come from the use of sub-pixel methods. These will certainly add a further computational burden to the process, and they may be avoided by using higher resolution and better quality images. In any case, the greatest contribution from this part of the work was the methodology that was developed for matching sets of points for which we have local contextual information in the form of edge and line contours: this information was treated as a “unary attribute” of the point, while the global context of the relative position of all points was incorporated with the help of probabilistic relaxation. This methodology may be used in solving other point matching problems where local and global context may be utilised.

## 6.2 Asymmetry Identification Using Generalised Symmetry Axis

The vast majority of facial and head trauma patients with isolated or severe injuries are injured as a result of road traffic accidents, falling, or beating. Internal bleeding, injuries and deformities in these kinds of patient destroy the symmetry, for example, of the frontal view of the face. In the second part of the thesis, we developed methodology for identifying the breaking of symmetry in CT images of parts of the human body, that are expected to be symmetric, and identifying the degree of asymmetry, as a diagnostic support tool for surgeons. We presented some generic methodology for defining a generalised symmetry axis in medical images. After detecting the symmetry axis, the difference between the two halves of the imaged organ were identified. The only assumption we made was that the axis was roughly vertical. This, however, is not particularly restrictive, as frontal imaging produces such images and in any case images may be rotated prior to using our algorithm to comply with this assumption. The method was applied to 1000 images and the detected

deformation areas agreed with the doctor's diagnosis, in each case. In this thesis, we did not make use of the 3D nature of the data. This generation of the methodology developed is not a straight forward process, but it may be used as the basis for a 3D extension. For a start, we shall have to deal with a surface of symmetry and not an axis of symmetry. The axes of symmetry for each slice identified by the method presented in this thesis may be combined to form the generalisation surface of symmetry, and the contours delineating the suspected deformities may be used to form 3D models of the suspect region to be viewed and manipulated in a 3D graphics package. The original ideas presented here, namely that of combining a gross global symmetry with local symmetries to detect local abnormalities, may also be used for other problems of anatomy detection.

# Bibliography

- [1] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *British Journal of Radiology*, 46:1016–1022, 1973.
- [2] S. Webb. *The Physics of medical imaging*. Bristol and Philadelphia: Adam Hilger, 1988.
- [3] K. J. Dreyer, D. S. Hirschorn, J. H. Thrall, and A. Mehta. *PACS : a guide to the digital revolution*. Springe, second edition, 2006.
- [4] R. M. Rangayyan. *Biomedical Image Analysis*. CRC Press, 2005.
- [5] H. C. Becker, W. J. Nettleton, P. H. Meyers, J. W. Sweeney, and C. M. Nice. Digital computer determination of a medical diagnostic index directly from chest x-ray images. *IEEE Transactions on Biomedical Engineering*, BME-11:67–72, 1964.
- [6] P. H. Myers, C. M. Nice, H. C. Becker, N. J. Nettleton, J. W. Sweeney, and G. R. Meckstroht. Automated computer analysis of radiographic images. *Radiology*, 83:1029–1034, 1964.
- [7] J. S. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:85–105, 2000.
- [8] B. Van Ginneken, B. M. Ter Haar Romeny, and M. A. Viergever. Computer-aided diagnosis in chest radiography: A survey. *IEEE transactions on medical imaging*, 20(12):1228–1241, 2001.

- [9] D. H. Ballard and J. Sklansky. Tumor detection in radiographs. *Computers and Biomedical Research*, 6:299–321, 1973.
- [10] J. Toriwaki, Y. Suenaga, T. Negoro, and T. Fukumura. Pattern recognition of chest x-ray images. *Computer Graphics and Image Processing*, 2:252–271, 1973.
- [11] M. Yanagisawa, S. Shigemitsu, and T. Akatsuka. Registration of locally distorted images by multiwindow pattern matching and displacement interpolation: The proposal of an algorithm and its application to digital subtraction angiography. *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1288–1291, 1984.
- [12] J. M. Fitzpatrick, J. J. Grefenstette, D. R. Pickens, M. Mazer, and J. M. Perry. A system for image registration in digital subtraction angiography. *Information Processing in Medical Imaging*, pages 415–434, 1987.
- [13] S. W. Zucker and R. A. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):324–331, 1981.
- [14] R. B. Schudy and D. H. Ballard. Model detection of cardiac chambers in ultrasound images. Technical report, Technical Report 12, Computer Science Department, University of Rochester, 1978.
- [15] R. Bajcsy, R. Lieberman, and M. Reivich. A computerized system for the elastic matching of deformed radiographic images to idealized atlas images. *Journal of Computer Assisted Tomography*, 7:618–625, 1983.
- [16] Niemann, H. Bunke, I. Hofmann, G. Sagerer, F. Wolf, and H. Feistel. A knowledge based system for analysis of gated blood pool studies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(3):246–259, 1985.
- [17] J. S. Duncan. Knowledge directed left ventricular boundary detection in equilibrium radionuclide angiocardiology. *IEEE Transactions on Medical Imaging*, 6(4):325–336, 1987.

- [18] J. Tsotsos. Knowledge organization and its role in representation and interpretation for time-varying data: The alven system. *Computational Intelligence*, 1(1):16–32, 1985.
- [19] C. Y. Han, K. N. Lin, W. G. Wee, R. M. Mintz, and D. T. Purembka. Knowledge-based image analysis for automated boundary extraction of transesophageal echocardiographic left ventricular images. *IEEE Transactions on Medical Imaging*, 10(4):602–610, 1991.
- [20] H. P. Chan, K. Doi, S. Galhotra, C. J. Vyborny, H. MacMahon, and P. M. Jokich. Image feature analysis and computer-aided diagnosis in digital radiography: Part 1 automated detection of microcalcifications in mammography. *Medical Physics*, 14:538–548, 1987.
- [21] M. O'Donnell, J. Gore, and W. Adams. Towards an automatic algorithm for nmr imaging ii: Initial segmentation algorithm. *Medical Physics*, 13:293–297, 1986.
- [22] D. A. Ortendahl and J. W. Carlson. Segmentation of magnetic resonance images using fuzzy clustering. In *Proceedings of Information Processing in Medical Imaging*, 39:91–106, 1988.
- [23] C. N. de Graaf, K. L. Vincken, M. A. Viergever, J. J. Koenderink, F. J. R., Appelman, and O. Ying-Lie. A hyperstack for the segmentation of 3d images. In D. A. Ortendahl and J. Llacer, editors, *Information Processing in Medical Imaging*, pages 399–413. Wiley-Liss, 1991.
- [24] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C. T. Chen. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. *Journal of Computer Assisted Tomography*, 13:20–26, 1989.
- [25] P. J. Besl and H. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–255, 1992.
- [26] L. H. Staib and J. S. Duncan. Model-based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, 15(5):720–731, 1996.

- [27] G. Szekely, A. Kalemien, C. Brechbuhler, and G. Gerig. Segmentation of 3D objects from MRI volume data using constrained elastic deformations of flexible fourier surface models. *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, pages 495–505, 1995.
- [28] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209, 1997.
- [29] P. C. Teo, G. Sapiro, and B. A. Wandell. Creating connected representations of cortical gray matter for functional MRI visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):852–863, 1997.
- [30] C. R. Meyer, J. L. Boes, B. Kim, P. Bland, K. R. Zasadny, P. V. Kison, K. Koral, K. A. Frey, and R. L. Wahl. Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin plate spline warped geometric deformations. *Medical Image Analysis*, 3:195–206, 1997.
- [31] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multi-modality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.
- [32] C. Nikou, B. Jaramaz, and A. M. DiGioia. Range of motion after total hip arthroplasty: Simulation of non-axisymmetric implants. *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 700–709, 1998.
- [33] C. Nikou, B. Jaramaz, A. M. DiGioia, M. Blackwell, M. E. Romesberg, and M. M. Green. POP: Preoperative planning and simulation software for total hip replacement surgery. *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 868–875, 1999.
- [34] B. H. Broadbent. A new x-ray technique and its application to orthodontia. *The Angel Orthodontist*, 1:45–46, 1931.

- [35] T. Rakosi. *An Atlas and Manual of Cephalometric Radiology*. Wolfe Medical Publications, 1982.
- [36] D. B. Forsyth, W. C. Shaw, and S. Richmond. Digital imaging of cephalometric radiography. 1. advantages and limitations of digital imaging. *The Angel Orthodontist*, 66(1):37–42, 1996.
- [37] I. El-Feghi, M. A. Sid-Ahmed, and M. Ahmadi. Automatic localization of craniofacial landmarks for assisted cephalometry. *Pattern Recognition*, 37:609–621, 2004.
- [38] V. Grau, M. Alcaniz, M. C. Juan, C. Monserrat, and C. Knoll. Automatic location of cephalometric landmarks. *Biomedical Information*, 34:146–156, 2001.
- [39] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [40] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image processing*, (2):266–277, February 2001.
- [41] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS, second edition, 1999.
- [42] B. Jahne. *Practical Handbook on Image Processing for Scientific and Technical Application*. CRC PRESS, second edition, 2004.
- [43] M. Petrou and P. Bosdogianni. *Image Processing, the Fundamentals*. WILEY, 2000.
- [44] C. Xu, A. Yezzi, Jr, and J. L. Prince. On the relationship between parametric and geometric active contours. *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems, and Computers*, 1:483–489, October 2000.
- [45] C. Xu and J. L. Prince. Snakes, shapes and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998.
- [46] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66(1):1–31, January 1993.

- [47] V. Caselles. Geometric models for active contours. *IEEE International Conference on Image Processing*, pages 9–12, October 1995.
- [48] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Fifth International Journal of Computer Vision*, pages 694–699, January 1995.
- [49] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. *Proceedings of the fifth IEEE International Conference on Computer Vision*, pages 840–845, June 1995.
- [50] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communication on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [51] S. Osher and J. A. Sathian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
- [52] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):266–280, March 2000.
- [53] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):755–268, June 2003.
- [54] J. A. Sethian. *Levelset methods and fast marching methods*. Cambridge University Press, second edition, 1999.
- [55] RM. Ricketts. Cephalometric analysis and synthesis. *The Angel Orthodontist*, 31:141–156, 1961.
- [56] C. Steiner. Cephalometrics in clinical practice. *The Angel Orthodontist*, 29:8–29, 1959.
- [57] R. Leonardi, D. Giordano, F. Maiorana, and C. Spampinato. Automatic cephalometric analysis, a systematic review. *The Angel Orthodontist*, 78(1):145–151, 2008.

- [58] S. Baumrind and R. Frantz. The reliability of head film measurements. 1. landmark identification. *American Journal of Orthodontics*, 60(2):111–127, 1971.
- [59] RC. Thurow. Cephalometric methods in research and private practice. *The Angel Orthodontist*, 21(2):104–116, 1951.
- [60] E. H. El-Mangoury, S. I. Shaheen, and Y. A. Mostafa. Landmark identification in computerized posterior-anterior cephalometrics. *American Journal of Orthodontics and Dentofacial Orthopedics*, 91:57–61, January 1987.
- [61] V. Ciesielski, A. Innes, S. John, and J. Mamutil. Genetic programming for landmark detection in cephalometric radiology images. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 7(3):164–171, 2003.
- [62] T. S. Douglas. Image processing for craniofacial landmark identification and measurement: a review of photogrammetry and cephalometry. *Computerized Medical Imaging and Graphics*, 28:401–409, 2004.
- [63] A. D. Levy-Mandel, A. N. Venetsanopoulos, and J. K. Tsotsos. Knowledge-based landmarking of cephalograms. *Computers and Biomedical Research*, 19:282–309, 1986.
- [64] L. Mero and Z. Vassy. A simplified and fast version of the huockel operator for finding optimal edges in pictures. *Proceedings of the fourth International Joint Conference on Artificial Intelligence*, pages 650–655, September 1975.
- [65] S. Parthasarathy, S. Nugent, P.G. Gregson, and D. F. Fay. Automatic landmarking of cephalograms. *Computers and Biomedical Research*, 22(3):248–269, 1989.
- [66] W. Tong, S. T. Nugent, P. H. Gregson, G. M. Jensen, and D. F. Fay. Landmarking of cephalograms using a microcomputer system. *Computers and Biomedical Research*, 23(4):358–379, 1990.
- [67] D. N. Davis and D. B. Forsyth. Knowledge-based cephalometric analysis: a comparison with clinicians using interactive computer methods. *Computers and Biomedical Research*, 27(3):210–228, 1994.

- [68] D. N. Davis and C. J. Taylor. A blackboard architecture for automating cephalometric analysis. *Medical Informatics*, 16(2):137–149, 1991.
- [69] C.K. Yan, A. N. Venetsanopoulos, and E. Filleray. An expert system for landmarking of cephalograms. *Proceedings of the Sixth International Workshop on Expert Systems and Applications*, pages 337–356, 1986.
- [70] P. H. Jackson, G. C. Dickson, and D. J. Birnie. Digital image processing of cephalometric radiographs: a preliminary report. *British Journal of Orthodontics*, 12(3):122–132, 1985.
- [71] A. M. Cohen and A. D. Linney. A low cost system for computer-based cephalometric analysis. *British Journal of Orthodontics*, 13(2):105–108, 1986.
- [72] A. M. Cohen, H. H. Ip, and A. D. Linney. A preliminary study of computer recognition and identification of skeletal landmarks as a new method of cephalometric analysis. *British Journal of Orthodontics*, 11(3):105–108, 1984.
- [73] J. Cardillo and M. A. Sid-Ahmed. An image processing system for locating craniofacial landmarks. *IEEE Transactions on Medical Imaging*, 13(2):275–289, June 1994.
- [74] M. Desvignes, B. Romaniuk, R. Demoment, M. Revenu, and M.J. Deshayes. Computer assisted landmarking of cephalometric radiographs. *Proceedings of the Fourth IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 296–300, 2000.
- [75] E. Uchino and T. Yamakawa. High speed fuzzy learning machine with guarantee of global minimum and its applications to chaotic system identification and medical image processing. *Proceedings of the Seventh International Conference Proceedings on Tools with Artificial Intelligence*, pages 242–249, November 1995.
- [76] Y. T. Chen, K. S. Cheng, and J. K. Liu. Improving cephalogram analysis through feature subimage extraction. *IEEE Engineering in Medicine and Biology Magazine*, 18(1):25–31, 1999.

- [77] A. Innes, V. Ciesielski, J. Mamutil, and S. John. Landmark detection for cephalometric radiology images using pulse coupled neural network. *Proceedings of the International Conference on Artificial Intelligence*, 2:511–517, June 2002.
- [78] T. Hutton, S. Cunningham, and P. Hammond. An evaluation of active shape models for the automatic identification of cephalometric landmarks. *European Journal of Orthodontics*, 22(5):499–508, October 2000.
- [79] D. J. Rudolph, P. M. Sinclair, and J. M. Coggins. Automatic computerized radiographic identification of cephalometric landmarks. *American Journal of Orthodontics and Dentofacial Orthopedics*, 113(2):173–179, February 1998.
- [80] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339, 1991.
- [81] T. F. Cootes, D. H. Cooper, C.J. Taylor, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [82] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Transaction on Systems, Man, and Cybernetics Society*, 6(6):420–433, 1976.
- [83] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*, volume 2. Academic Press, Inc, 1982.
- [84] S. Peleg and A. Rosenfeld. A note on the evaluation of probabilistic labelings. *IEEE Transaction on Systems, Man, and Cybernetics Society*, 11(2):176–179, 1981.
- [85] B. Bhanu and O. D. Faugeras. Segmentation of images having unimodal distributions. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 4(4):408–419, 1982.
- [86] A. J. Danker and A. Rosenfeld. Strip detection using relaxation. *Pattern Recognition*, 12:97–100, 1980.

- [87] M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142–159, 2000.
- [88] H. Zhao, K. Chan, L. Cheng, and H. Yan. A probabilistic relaxation labeling framework for reducing the noise effect in geometric biclustering of gene expression data. *Pattern Recognition*, 42:2578–2588, 2009.
- [89] H. Wang and E. R. Hancock. Probabilistic relaxation labelling using the fokkerplanck equation. *Pattern Recognition*, 41:3393–3411, 2008.
- [90] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.
- [91] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *In Proceedings of the IEEE International Conference on Cybernetics and Society*, pages 163–165, 1975.
- [92] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [93] H. Weyl. *Symmetry*. Princeton University Press, 1952.
- [94] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transaction on Pattern analysis and Machine Intelligence*, 17(12):1154–1166, 1995.
- [95] K. Kanatani. Comments on “symmetry as a continuous feature”. *IEEE Transaction on Pattern analysis and Machine Intelligence*, 19(3):246–247, 1997.
- [96] E. Yodogawa. Symmetropy, an entropy-like measure of visual symmetry. *Perception and Psychophysics*, 32(3):230–240, 1982.
- [97] G. Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):104–108, 1989.

- [98] S. L. Chou, J. C. Lin, and W. H. Tsai. Fold principal axis: a new tool for defining the orientations of rotationally symmetric shapes. *Pattern Recognition Letters*, 12(2):109–115, 1991.
- [99] C. Sun and D. Si. Fast reflectional symmetry detection using orientation histograms. *Real-Time Imaging*, 5(1):63–74, 1999.
- [100] Y. Keller and Y. Shkolnisky. An algebraic approach to symmetry detection. *Proceedings of the 17th International Conference on Pattern Recognition*, 3:186–189, 2004.
- [101] H. Blum and R.N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.
- [102] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.
- [103] R. A. Brooks. Symbolic reasoning among 3d models and 2d images. *Artificial Intelligence*, 17:285–348, 1981.
- [104] J. Bigün. Recognition of local symmetries in gray value images by harmonic functions. *The 9th International Conference on Pattern Recognition*, 1:345–347, November 1988.
- [105] J. Ponce. On characterizing ribbons and finding skewed symmetries. *Computer Vision, Graphics, and Image Processing*, 52:328–340, 1990.
- [106] D. Reifeld, H. Wolfson, and Y. Yeshurun. Detection of interest points using symmetry. *Proceedings of the Third International Conference on Computer Vision*, pages 62–65.
- [107] K. Cho and S. M. Dunn. Hierarchical local symmetries. *Pattern Recognition Letters*, 12:343–347, 1991.
- [108] S. Mitra and Y. Liu. Local facial asymmetry for expression classification. *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, 2:889–894, 2004.

- [109] T. Zielke, M. Brauckmann, and W. von Seelen. Intensity and edge-based symmetry detection with an application to car-following. *CVGIP: Image Understanding*, 58(2):177–190, 1993.
- [110] A. V. Tuzikov, O. Colliot, and I. Bloch. Evaluation of the symmetry plane in 3d mr brain images. *Pattern Recognition Letters*, 24:2219–2233, 2003.
- [111] M. Mancas, B. Gosselin, and B. Macq. Fast and automatic tumoral area localisation using symmetry. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2:725–728, 2005.
- [112] Y. Bonnef, D. Reisfeld, and Y. Yeshurun. Texture discrimination by local generalized symmetry. *Proceedings of Fourth International Conference on Computer Vision*, pages 461–465, May 1993.
- [113] D. Chetverikov. Pattern orientation and texture symmetry. pages 222–229, 1995.
- [114] R. Alterson and D. B. Plewes. Bilateral symmetry analysis of breast mri. *Physics in Medicine and Biology*, 48:3431–3443, 2003.
- [115] L. Junck, J. G. Moen, G. D. Hutchins, M. B. Brown, and D. E. Kuhl. Correlation methods for the centering, rotation, and alignment of functional brain images. *The Journal of Nuclear Medicine*, 31:1220–1226, 1990.
- [116] S. Joshi, P. Lorenzen, G. Gerig, and E. Bullitt. Structural and radiometric asymmetry in brain images. *Mathematical Methods in Biomedical Image Analysis*, 7:155–170, 2003.
- [117] S. Prima, S. Ourselin, and N. Ayache. Computation of the mid-sagittal plane in 3-d brain images. *IEEE Transactions on Medical Imaging*, 21:122–138, 2002.
- [118] H. Blum. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [119] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–587, 1973.

- [120] M. Leyton. A process-grammar for shape. *Artificial intelligence*, 34(2):213–247, 1988.
- [121] D. P. Mukherjee, A. Zisserman, and M. Brady. Shape from symmetry - detecting and exploiting symmetry in affine images. *Philosophical Transactions of the Royal Society of London*, 351(1695):77–106, 1995.
- [122] M. J. Atallah. On symmetry detection. *IEEE Transactions on Computers*, pages 663–666, 1985.
- [123] J. Sato and R. Cipolla. Affine integral invariants for extracting symmetry axes. *Image and Vision Computing*, 18(8):627–635, 1997.
- [124] A.M. Bruckstein and D. Shaked. Skew symmetry detection via invariant signatures. *Pattern Recognition*, 31(2):181–192, 1998.
- [125] Y. Gofman and N. Kiryati. Detecting symmetry in grey level images: the global optimization approach. *Proceedings of the 13th International Conference on Pattern Recognition*, 1:889–894, 1996.
- [126] V. Di Gesu and C. Valenti. Symmetry operators in computer vision. *Vistas Astron*, 40(4):461–468, 1996.
- [127] V. Di Gesu and C. Valenti. Detection of regions of interest via the pyramid discrete symmetry transform. In *Advances in Computer Vision* (Solina, Kropatsch, Klette, and Bajcsy editors), Springer-Verlag, 1997.
- [128] R. Manmatha and H. Sawhney. Finding symmetry in intensity images. *Technical Report*, 1997.
- [129] D. Shen, H.H.S. Ip, and E.K. Teoh. An energy of asymmetry for accurate detection of global reection axes. *Image and Vision Computing*, 19(5):283–297, 2001.
- [130] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
- [131] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

- 
- [132] S. Periaswamy and H. Faridb. Medical image registration with partial data. *Medical Image Analysis*, 10(3):452–464, 2006.

## Appendix A

# The Cephalometric Landmarks And Their Meaning

- **S** (Sella): midpoint of sella turcica, a constructed radiologic point in the median plane
- **N** (Nasion): junction of the frontal and nasal bones at the naso-frontal suture (suture: the line of union of two bones)
- **ORB** (Orbitale): most inferior point on the infraorbital rim
- **PTMS** (Pterygoid): landmark of the posterior shadow of pterygomaxillary fissure, a bilateral teardrop-shaped area of radiolucency, the posterior of the shadow which represents the pterygoid plates, whereas the anterior shadow of which represents the posterior surfaces of the tuberosities of the maxilla
- **PTM**(Pterygomaxillary fissure): landmark at the 6 o'clock position of the midplaned contour of the pterygomaxillary fissure, at the junction of the pterygoid plates and the maxilla
- **PNS** (Posterior nasal spine): posterior limit of the floor of the nose, at the tip of the posterior nasal spine
- **POR** (Porion): the most superior point of the external auditory canal

- **CO** (Condylion): the most posterosuperior point of the mid-planed contour of the mandibular condyle
- **ART** (Articulare): a mid-planed point located at the intersection of the posterior border of the ramus with the inferior surface of the cranial base
- **BA** (Basion): most inferior point on the anterior margin of foramen magnum, at the base of the clivus
- **ANS** (Anterior nasal spine): the anterior limit of the floor of the nose, at the tip of the anterior nasal spine
- **A** (Point A): the deepest point in the concavity of the anterior maxilla between the anterior nasal spine and the alveolar crest
- **PR** (Prosthion): alveolar rim of the maxilla; the lowest, most anterior point on the alveolar portion of the premaxilla, in the median plane, between the central incisors
- **LIE** (Lower incisor tip): the tip of the crown of the lower central incisor
- **UIE** (Upper incisor tip): the tip of the crown of the upper central incisor
- **ID** (Infradental): the highest, most anterior point on the alveolar process, in the median plane, between the mandibular central incisors
- **B** (Point B): the deepest point in the curvature of the mandibular alveolar process
- **POG** (Pogonion): the most anterior point on the bony chin
- **GN** (Gnathion): the midpoint between pogonion and menton
- **ME** (Menton): the extreme inferior point of the chin
- **LIA** (Lower incisor apex): the root apex of the lower central incisor
- **GO** (Gonion): the midpoint of the mandibular angle between ramus and corpus mandibulae

## Appendix B

# Algorithm for Local Histogram Equalization

The size of the sliding window is  $(2m + 1) \times (2m + 1)$

Every sub-image inside the window is treated like a full image, and histogram equalization is applied to it. Only the pixel in the center of the window is modified. Then the window is shifted by a pixel. This process is repeated for every pixel. The algorithm is as follows:

Create an output array the same size as the input image and set all its values to 255.

Initialize the size of the window.

Visit all pixels one after the other.

**do**

**if** the pixel in the center of the sliding window is not flagged as bone region in the mask image, **then**

do nothing.

**else**

Put all neighbours of the central pixel that are flagged as bone in an one dimensional array.

Put the central pixel at the end of the array.

Apply histogram equalization to that array.

Take only the last value of the resultant array (the modified central pixel) as an output value and place it in the corresponding position of the output image.

**end if**

Shift the sliding window by a pixel.

Repeat this process **until** the end of the image.