AUTOMATIC G^1 PARAMETRIC FITTING OF CURVES AND SURFACES TO OUTLINES OF IMAGES

FATIMAH YAHYA

UNIVERSITI SAINS MALAYSIA

2009

AUTOMATIC G^1 PARAMETRIC FITTING OF CURVES AND SURFACES TO OUTLINES OF IMAGES

by

FATIMAH YAHYA

Thesis submitted in fulfillment of the
requirements for the degree of
Doctor of Philosophy

November 2009

Acknowledgements

In the name of Allah the most beneficent, most merciful.

I would like to express my thanks and gratitude to all the people that are instrumental in making this thesis a reality.

First and foremost, I would like to thank my supervisors, Associate Professor Dr. Jamaludin Md. Ali, Associate Professor Dr. Ahmad Abdul Majid and Associate Professor Dr. Arsmah Ibrahim for their guidance, their many ideas, comments and valuable advice. I am forever grateful for their inspiration, encouragement and support for this undertaking.

I am also thankful to the Faculty of Computer and Mathematical Sciences, UiTM, University Technology MARA, Jabatan Perkhidmatan Awam and the government of Malaysia for giving me the opportunity and the finance to embark on this study. My thanks also to the Pusat Pengajian Sains Matematik, Universiti Sains Malaysia for providing a conducive environment of education and research.

Lastly, I wish to record my appreciation for the wholesome support of my extended family especially my parents; mother Maimun Mohamed and my dear departed father Yahya Haji Salleh; my siblings, nephews and nieces. Thanks for all their prayers.

Praise be to Allah for giving me time and strength to finish this endeavour.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PENYESUAIAN LENGKUNG DAN PERMUKAAN G^1 SECARA AUTOMATIK PADA BENTUK LUARAN IMEJ

ABSTRAK

Perkembangan pesat teknologi pengimejan menghasilkan sejumlah besar data yag boleh dipergunakan bagi pengumpulan maklumat dan pengetahuan. Perwakilan bermatematik objek di bawah perhatian dari imej-imej, boleh dimanipulasi bentuk dan saiznya. Ini membantu dalam penganalisisan dan rekabentuk.

Sebagai suatu proses kejuruteraan undur, sasaran adalah untuk menghasilkan bentuk luaran bermatematik, imej kontur 2-D sesuatu objek secara automatik. Kemudian pembinaan semula suatu objek atau permukaan 3-D dari imej-imej keratan rentas akan dihasilkan, juga secara automatik. Antara objektifnya adalah untuk mendapat suatu perwakilan yang boleh dipercayai, cepat dan mempunyai ketepatan yang fleksibel.

Kebanyakan pendekatan kepada pembinaan semula daripada kontur, menghasilkan permukaan C^0 atau C^1 dengan jejaring tiga segi. Teknik kami menghasilkan suatu lengkung bersesuaian kontur sesatah dan seterusnya permukaan G^1 daripada siri kontur-kontur yang memuaskan secara geometri.

Mula-mula sekali, bentuk luaran kawasan di bawah perhatian diambil. Dengan menggunakan nilai-eigen matriks kovarian dan konsep lingkungan sokongan , titik penjuru dikesan. Titik-titik penjuru ini adalah dalam pelbagai darjah kelicinan.

Penyesuaian lengkung dilakukan dengan penyisipan titik-titik penjuru dan penghampiran data lain yang tinggal. Kubik-kubik Bezier nisbah G^1 yang ditentukan secara lelaran, disesuaikan antara titik-titik penting ini. Pemberat diubahsuai secara automatik untuk mendapat lengkung terdekat kepada titik-titik data digital seperti dikehendaki. Suatu parameter global panjang lengkuk dihampiri pada ketepatan ditentukan untuk setiap splin di setiap kontur. Lengkung kontur bersebelahan kemudian digabung untuk membina permukaan.

Kejituan permukaan dikawal oleh suatu ukuran toleransi. Toleransi tinggi memberi suatu imej kasar manakala toleransi rendah menghasilkan imej lebih halus. Imej halus ini akan memerlukan pengiraan yang lebih. Rupa-bentuk boleh dikenalpasti daripada data titik-titik penjuru dan diasingkan. Oleh itu, beberapa bahagian daripada keseluruhan permukaan boleh dihasilkan secara berasingan. Teknik ini adalah automatik sepenuhnya.

AUTOMATIC G^1 PARAMETRIC FITTING OF CURVES AND SURFACES TO OUTLINES OF IMAGES

ABSTRACT

Rapid advancement in imaging technologies produces massive amount of data which can be harnessed for information and knowledge gathering. Mathematical representations of objects of interest from these images are amenable to manipulation of shapes and sizes, thus aiding analysis and design.

As a process in reverse engineering, we aim to automatically reproduce a mathematical outline of a 2D contour based image of an object. Next we will reconstruct a 3D object (surface) from its cross-sectional images. It is our objective to have a representation which is reliable, reasonably fast and with flexible accuracy.

Most existing approaches to shape reconstruction from contours, yield C^0 or C^1 surfaces with triangular meshes. Our technique produce a geometrically pleasing, G^1 curve fitting of planar contours and a G^1 surface from these series of contours.

First the outlines of the region of interest are extracted from the 2D image. Then using eigenvalues of covariance matrix and employing the concept of region of support, corner points of the outlines which are of varying degrees of smoothness are detected. Curve fitting is done by interpolating corner points and approximating the rest of the data. G^1 rational Bezier cubics, iteratively determined, are fitted piecewise between these significant points. The weights are adjusted automatically to get curves that are as close as needed to the digitized data points. A global arc length parameter is approximated at specified accuracy, for the spline at each contour. Adjacent contour curves are then blended together to form the surface.

Surface accuracy is controlled by a tolerance measure. A high tolerance gives a coarse image while low tolerance produces finer images, albeit with more computation. Features can be identified from the corner points data and isolated. Thus, certain parts of the whole surface can be produced separately. Our technique is fully automatic.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In engineering, medical sciences, digital art and other computer-based applications, there is a great need to create a computer representation of existing objects from some measured data. Most times, this representation need to be an accurate, consistent model of the original as it is used for numerically controlled precise manufacturing, for further designing and analysis, or to be used in verification and identification purposes such as in 3D face recognition and facial expression simulation and in facilitating medical procedures. It could also be simply an efficient method of storing visual data taking up drastically reduced storage space.

Medical industry and services produce copious amount of images through its imaging facilities like computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET) and confocal microscopy. The acquired scans are usually segmentally parallel sections. A 2D or 3D reconstruction of explicit models of interest, generate better understanding of its shape, structure and geometrical characteristics which are a great help in diagnostic considerations and surgical preparations. It is also a great aid in biological research.

Recent advances in scanning devise like the laser scanners easily produce real 3D data. Also called range data, it gives geometric and texture information of an object. This type of data can also be utilized in reconstructing surfaces of solids or models.

A 2D and 3D mathematical model will render itself well to manipulation of shape and size which is most applicable to computer-based applications like plastic surgery, animation and data verification. In face recognition and medical use for example, reliably accurate facial models are required. For animation or digital art, image details are not really wanted but speed of data transfer is of importance. Thus a coarser model is sufficient. Variation in accuracy of such models is an advantage.

1.2 Objectives

The first objective of this thesis is to automatically reproduce the outline of contour-based images in 2D. As a method in reverse engineering, it is created by way of digitizing an image that already exists and then fitting G^1 curves automatically to the outline of the digitized images. These planar reproductions will be applied to font and medical images.

Arabic font is chosen as it is difficult to fit because of its cursive character, having varying curvatures and cusps. Contours of anatomy or tumors bore certain similarities with outline shape of Arabic characters. Both are mostly circular based, containing smooth curves and corners. The work on Arabic fonts is extended to medical images.

A second objective is to build up an automatic mathematical parametric 3D representation of a face from a series of planar contours. What is needed is a reliable reproduction of a particular individual’s face and not a generic one obtained from a few measured data. The representation should also be readily manipulated in shape and size. This work will try to visualize the 3D surface of a face (notably Asian) or a particular anatomy of a face, as an effort of reconstruction after deformation.

Being automatic, it releases doctors and researchers from doing repetitive manual delineation. But more importantly, it furthers the depth of analysis whereby the object under study now has a definable contour or boundary. Analysis of growth, effects of tumour growth on its surrounding tissue or bone and visualizations of deformation and reconstruction of anatomy before and after surgery are some of its possible uses.

It is also our objective to not only have a reliable representation, but also a fast one and easily accessible. Some medical softwares like 3D Slicer need a dedicated workstation for its 3D visualizations. The program would just stop responding on a common laptop. Our algorithm will be built in Mathematica on a 2-CPU laptop with 2 GB RAM.

1.3 Review of Techniques

There are 3 major ways in creating 3D models from 2D images; volume visualizations, polygonal modelling and using parametric patches. In volume visualizations, the sequence of 2D images are stacked on top of one another and a 3D matrix of voxels is created by raster interpolation of adjacent images along the z-axis. The object of interest in the volume data set can be extracted by 2D or 3D segmentation procedures. For further refinement of the rugged boundary, the marching cubes algorithm of Nielson (2003) is usually used to construct a polyhedral model for representation of the object. This high resolution surface model comprises many tiny triangles. Algorithms of this category give good results but are very time consuming.

In the polygonal modelling approach, the outlines of structure of interest are obtained at each slice. These outlines or contours are stacked and a polygonal mesh is created between successive contours by connecting vertices or points on one contour to vertices (points) on the other. Meshes are either of triangular or rectangular form. This gives rise to the problem of corresspondence (how to connect vertices between contours), tiling (how to create meshes from edges) and branching (how to cope when there are slices with different number of contours).

Polygonal or surface models and parametric patches are more difficult to construct initially, but they are faster, flexibly scaled, easily manipulated and usually are a more compact representation of the structure. They are also compatible with surface-based graphical tools available today and readily imported into CAD/CAM programs for further designing or manufacturing purposes.

Most existing approaches to shape reconstruction yield C^0 triangular surfaces where each triangular face is made up of 2 consecutive vertices from one contour to one vertex of the other contour. Fuchs et al.(1997) define the best reconstructed surfaces as the one with minimal surface area while Keppel (1975) bounds the maximum volume. Bajaj et al. (1996) provide a unified approach to solving the correspondence, tiling and branching problem by imposing three constraints on the surface when reconstructing. Stewart and McCracken (2002) discuss a semi automated system which uses neural network in the region delineation task with triangulation between contours.

Fujimura and Kuo (1999) use isotropic deformation to create non self intersecting surfaces from nested contours citing that triangular tiling may at times gives rise to a visual artifact in the reconstructed shapes due to discontinuity in surface normal. Such discontinuity is less noticeable for rectangular tiling or free form surfaces. Triangular tilings which is piecewise linear also makes it difficult to incorporate feature corresspondence. Chen et al. (2007) give a correspondence determining algorithm followed by a hybrid tiling algorithm to tile contours.

Johnstone and Sloan (1995) describe a method for creating Bezier surfaces from contours with cylindrical properties. Song et al. (2006) use B-spline surface patches to reconstruct human faces from 3D point cloud data. Eyad and Hassan (2007) segment the facial range data into several patches that is deemed to be adequately representing the surface. They then reconstructed the human face with partial differential equation approximating the surface between 4 boundary curves for each patch.

Chen et al.'s (2008) segmentation of confocal microscopy images is an example of 3D reconstruction in biomedicine. They use a template driven technique to produce 3D models of anatomic structure from series of images. Firstly, employing 3D segmentation method a template polyhedral model is produced. This model is then sliced giving 2D contours which acts as initial contours for the active snake contour models. The next step is contour refinement which is done iteratively from slice to slice until the difference between 2 successive iterations reach a specified tolerance or the number of iterations exceed a specified number. These final contours are used to reconstruct the polyhedral model of the anatomy concerned.

Mishchenko (2009) meanwhile, reconstruct neural tissue for large number of serial sections micrographs by first detecting the cell profiles of each image in the series with a multi-scale ridge detector. The adjacent profiles are then linked based on their shape and texture producing a 3D segmentation.

If the serial profiles of object is mathematically represented the 3D representation can be made to be amenable to manipulations and distortion. This is very useful in facial reconstruction surgeries and animation.

Sarfraz (2002 – 2008) extract outlines of images and then approximately profiles them with parametric bezier spline curves. The approximation process involves getting boundary of the planar object, detecting corner points and fitting curves to this boundary through the corner points and other additional points or knots when necessary to achieve good approximation. Sarfraz (2002) uses least square fitting to approximate C^1 cubic spline to the corner points (or points of high curvature). The approximation is made better by splitting the curve segment at points where the distance between the curve and the boundary is the greatest, and refitting it. Sarfraz (2007a) and Masood (2008), do not use the expensive least square fitting. In Sarfraz (2007a), the unknown middle control points of the Bezier cubic is obtained by exploiting the properties of a bezier curve. An optimal approximation is found by moving the initial middle control points along its slope. Splitting and refitting is done to achieve accepted threshold. Masood (2008) also obtained the control points mathematically from the given curve. The technique produces a control points spread or CPspread where if the spread does not satisfy a threshold value, the curve will be subdivided. The technique is fast but since it does not approximate the error, the accuracy of the fit in absolute terms is not known, but is reflected in the CPspread.

Sarfraz (2007b) fit piecewise parametric cubic spline interpolant with shape parameters, between corner points. These parameters tightened or loosened the curves to fit the data. The optimal fit is obtained by optimizing the parameters using Stochastic Evolution technique developed by Saab (1991). Sarfraz (2008), use an iterative simulated annealing technique in adjusting shape parameters to optimally fit a G^1 cubic spline to the segments formed by the corners and knots. Sarfraz

(2007c) obtain the knot points using a fuzzy randomized knot insertion technique. A cubic spline interpolant is fitted through all the corner and knot points. The approximation accuracy of spline to the outline is measured by the distance of a chosen random point to the digitized outline. Thus much of the computation is reduced.

The review for the particular techniques of boundary extraction, corner detection and arc-length re-parameterization will be dealt with in their particular chapters.

1.4 Work scheme

It is the aim of this work to fit geometric curves to the contours of digitized images. The whole process should be automatic. The work is started off with fitting of Arabic fonts that could preserve much of its cursive, smooth flowing calligraphic character. Arabic fonts are chosen for they contain varying geometric forms, i.e. curves of varying degree of smoothness, cusps, inflections and straight sections. Successful fitting of these characters bode well for fitting of any other images.

The work is then extended to finding outlines of medical images which are very useful. Arabic fonts and medical images of anatomy or growth are similar in the sense that they are circular based, containing smooth curves and corners. There is actually extensive literature on finding the outlines.

The method consists of two main parts- segmentation and curve fitting. Segmentation involves digitizing the image, getting its outline or boundary, and then obtaining equidistant data points to represent the outline. Using a suitable corner detection method, the corners or corner data points of the outlines are identified. This first stage breaks the contour outline into segments at the corners.

The next stage fit curves to these segments. A good fitting or representation of the outline will depend much on good determination of corner points. Spurious corner points cause more segmentation than necessary whilst undetected corner points hamper optimum fitting which may again require additional segments. More segments not only mean more computation but also a representation which is more fragmented and less smooth.

The fitted curve should ideally be smooth and reproducing outlines which are accurate to the image it represents. The curves will be approximating the contour data points and only interpolates some - primarily the corner points. Goodness of fit will be measured by how close the curves are to the contour data points. The curves need to be within a certain tolerance to them. This tolerance is the maximum distance between the curve and any contour data points.

The curves are made to be geometrically smooth at joints. Mathematical smoothness is not preferred as loops may form in order to maintain this type of continuity at joints which is geometrically not pleasing.

The fitting is started off with G^1 Bezier cubics. If the fitted curves does not satisfy the tolerance, the curve is framed in rational cubics. Automatic adjustments and determination of these weights using an iterative technique are employed to reach the tolerance. If the iterative limit is reached but the tolerance is still not satisfied, the curve is then split and refitted at additional points. The whole fitting process is repeated until the tolerance is reached. Thus the curves interpolate 'significant points' –corners and the additional points- and approximate other boundary points to a certain specified tolerance. The curves are G^1 composite rational Bezier cubics.

The serial contours are stacked and blended together to form the surface. To do this, corresponding points on the contours are established and matched. This is done by reparameterizing the contours in arc-length parameter. Since the analytical arc length reparameterization is almost impossible to do, an approximate tailored to the required accuracy will be obtained. A fast technique using cubic Bezier is used to parameterize the contour as functions of arc length.

The surface is formed when the contours are interpolated using monotonic cubic Hermite interpolation, producing rectangular meshes.

Our technique produces a G^1 surface from cross sectional contour curves of images. Surface accuracy is controlled by a tolerance measure. A high tolerance gives a coarse image while a low tolerance produces finer, more accurate images, albeit with more computations. Features can be identified from the significant points data and isolated. Thus certain parts of the whole surface can be produced separately; for example the nose or ears from a face. The technique is fully automatic except for determining the first boundary point of the first image cross section. See Figure1.1 for the full algorithm.

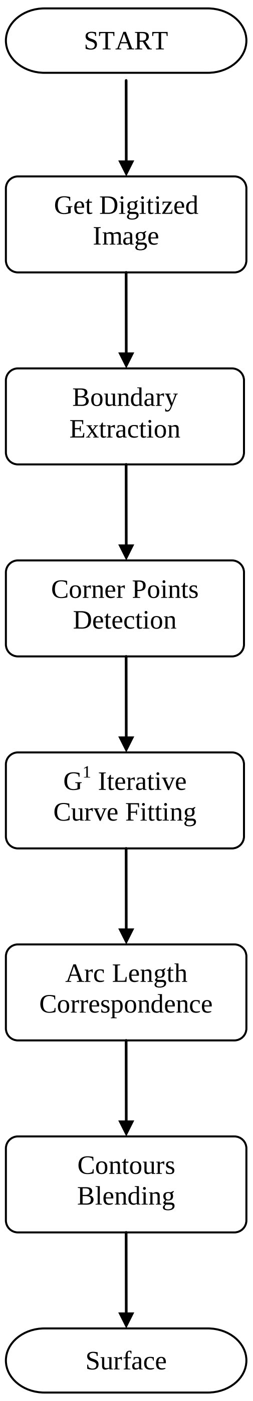

Figure 1.1 The main algorithm

All programming and rendering are done in Mathematica. The image of the Arabic font is scanned into the computer while Slicer-3D is used to access the computed tomography, CT images and for region thresholding. The 3D visualization technique is applied to a set of 120 dicom images of the head at 1.5mm apart to visualize a human face. All work is done on a 1.83 GHz, 2 GB RAM computer laptop.

1.5 Outline of thesis

Chapter 1 states the motivation and objectives of the study. It also contains a review on the major ways of reconstructing 3D surfaces from 2D planar images and a scheme of work which details the methodology and main algorithm.

Chapter 2 starts with the first step in the main algorithm which is boundary extractions. The next step of corner detection is addressed in Chapter 3. It consists of a review of corner detection techniques, introduces and explains our technique and the subsequent results when applied to Arabic font images.

Chapter 4 dwells on parameterization of the fitting curve and how reparameterization can be done. Chapter 5 explains the fitting of G^1 rational cubic Bezier between the corner points detected. The outline of the process is given in Figure.5.2. The technique is again applied to font images and the results are discussed.

Arc length reparameterization is dealt with in Chapter 6. A review of present techniques of arc length reparameterization can be found in the introductory section. The technique used in this thesis is presented and explained. The performance of the technique is evaluated on different shaped curves. As each contour of the planar images consists of many cubic rational curves composed together, a global parameter is necessary to sew them up. This is done at the end of Chapter 6. Chapter 7 deals with Hermite blending of the contours.

Chapter 8 discusses the results obtained by applying the algorithm to serial CT images of the head producing the facial image of a person. Chapter 9 concludes the thesis by giving an overview of the whole process of reconstruction and its results.

CHAPTER 2

CONTOUR EXTRACTIONS

Digital images of the Arabic letters are obtained by scanning its designed pictures. The CT scanned images are accessed through Slicer 3D and saved as bitmap images in the computer to be processed. The boundary is then extracted. There are many ways to get the boundary, such as in Avrahami and Pratt (1991), Gonzalez and Woods (2002) and other good edge detection techniques. Some involve getting it directly from gray level images to minimize errors in detection. However getting the boundary from binary image is simpler and faster. The font image here is considered a "simple" image i.e. without tremendous detail since it is two tone - black characters on white background, justifying the use of binary methods.

However care is taken to lessen errors in detection. Preprocessing should be done before converting into binary image by thresholding. A closed contour of one pixel thick is extracted from the binary image.

2.1 Contour extraction

In this thesis the digital images obtained from scanning is smoothed out using a 7 by 7 averaging spatial filter. The resulting digital image is then converted into binary image by thresholding. The boundary of the image is then extracted using a technique in mathematical morphology, $\beta(A) = A - (A \ominus B)$ where A is the set of all black pixels, B is the 3x3 structuring element and $\beta(A)$ is the boundary of set A. The operations Θ and - represent the operation of erosion and difference respectively.

The chosen single-pixel, digitized data-points to represent the contours of the images are obtained by using a 7 by 7 spatial filter. This filter is made to move along the boundary, extracting the position of every third pixel on the boundary and throwing away, the intermediate two, as demonstrated in Figure 2.3. The data-points obtained are already ordered, either clockwise or anti-clockwise as desired.

If the data-points of the contour are too close together, some points can be thrown out from the clockwise (anti-clockwise) arrangement. The use of larger spatial filter (in the previous paragraph) to get wider-spaced data-points is not advisable as it may not be able to retain contour details especially at sharp corners. The products are contours made of data points that are approximately equidistant from each other. Here the distance is approximately three pixels.

2.2 Contours of font images

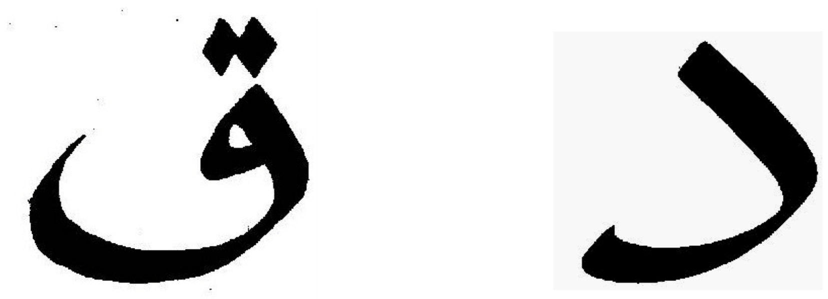
a. b.

Figure 2.1 Digitized images of a. “qaf” and b. “dal”

The digitized images of the Arabic font "qaf" and "dal" in Figure 2.1 are produced by scanning their designed images. The outlines of the fonts as in Figure 2.2 are then obtained by mathematical morphology. These outlines are later replaced by equidistant data points obtained by the technique explained in Section 2.1. The data points are approximately three pixels apart as shown in Figure 2.3.

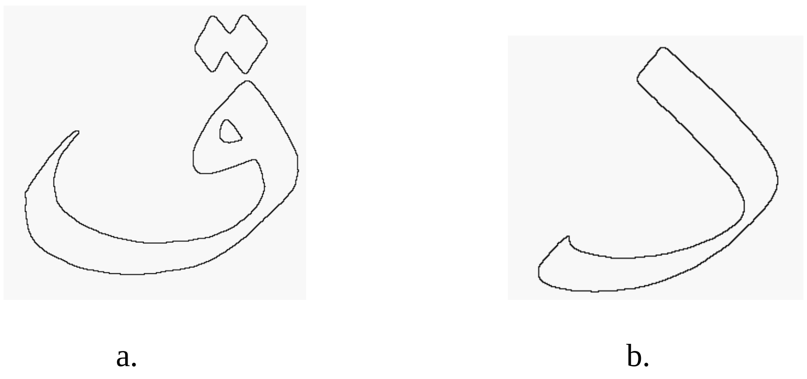

Figure 2.2 Outlines of a. "qaf" and b. "dal"

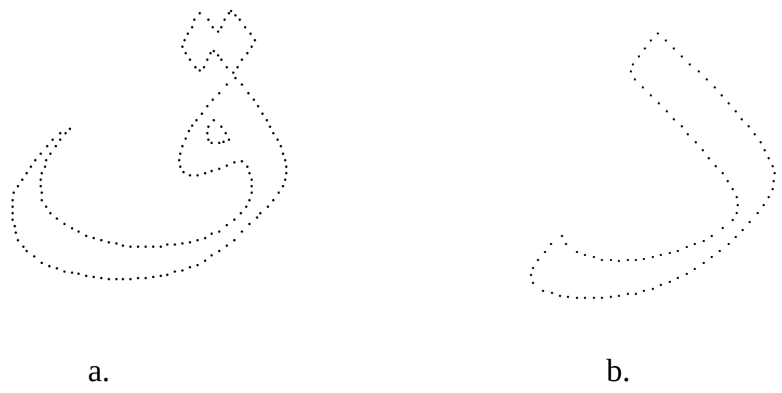

Figure 2.3 Contour of a. "qaf" and b. "dal"

The result for the boundary extraction of CT scan images will be dealt with later together with its 3D visualization.

CHAPTER 3

CORNER DETECTION

3.1 Review

Corner detection methods fall broadly into 2 categories – ones that act directly on gray levels and those that are based on boundaries. There are two main techniques in gray level methods. Template-based corner detection involves determining similarities between templates of specific angles with all sub-windows of the same size in the image. This technique need a huge amount of computation. Gradient-based corner detection measures the curvature of an edge that passes through a neighborhood of the gray level image by using the edge strength and gradient of edge direction. This method is not good at localizing corners.

Methods based on boundaries are simpler and faster. Boundaries are represented either by points or chain codes. Generally chain code methods like Freeman and Davis (1977), Rosenfeld and Johnson (1973), Rosenfeld and Weszka (1975), Beus and Tiu (1987), Rutkowski and Rosenfeld (1978), Cheng and Hsu (1988), Sankar and Sharma (1978) and Medioni and Yasumoto (1987) involve computing some measure or estimate of curvature or measure of significance at each point. The corner points are those that are above a certain threshold and are a local maximum. Most of these techniques are dependant on an input parameter which actually refers to the degree of smoothing used when calculating the estimates of curvature. The Rosenfeld-Johnson (1973) corner detector is the *k*-cosine. For point *i* with coordinates (x_i, y_i), the *k*-cosine (c_{ik}) is given as

$$c_{ik} = \frac{a_{ik} \bullet b_{ik}}{|a_{ik}||b_{ik}|} \quad where \quad a_{ik} = (x_i - x_{i+k} \; , \; y_i - y_{i+k}) \; , \qquad b_{ik} = (x_i - x_{i-k} \; , \; y_i - y_{i-k})$$

and k is the variable degree of smoothing. Corner points are those with c_{ik} greater than a certain threshold and are local maxima.

A large degree of smoothing means that the image is looked upon at a big scale thus losing the finer details or features of the curve. On the other hand, a small degree of smoothing may give too much detail of which some may be spurious. On a digital image which has multiple size features, a large degree of smoothing may result in undetected dominant or corner points, whereas a small degree may result in false corners being present. Sankar and Sharma’s (1978) algorithm do not require input parameters and takes into account various levels of detail but it is not effective for image with different size corners. The problem of corner detection is not just about getting an accurate estimate of curvature but also a matter of scale or alternatively the localization of corners.

Witkin (1983) introduced scale-space analysis which deals with descriptions of the digital boundaries at multiple scales, integrating and managing them to get worthwhile information. Mokhtarian and Mackworth (1986) further applied it to two-dimensional shapes, locating zero-crossings of curvatures. Lindeberg (1990) developed the scale-space theory for discrete signals. Scale-space analysis of a signal is generally made by convolving it with the Gaussian kernel treating its parameter as a continuous scale parameter. A scale-space map is produced where the arc length is shown along the x-axis and the scale parameter along the y-axis. The image in this half-plane shows location of curvature extreme or zero crossings of curvature at varying scales. Rattarangsi and Chin (1992), Ray and Ray (1997) and Asada and Brady (1986) detect corners using scale-space, but these methods are computationally heavy and the dominant points need to be tracked down from the scale-space map. The digital scale-space map is organized to a well-defined structure of a tree and later parsed to detect corners or dominant points. Saint-Marc et al (1991) present an adaptive smoothing method to produce a new scale-space representation without using the Gaussian kernel. The method iteratively convolves the signal with a small averaging mask weighted by a measure of the signal continuity at each point. No tracking of dominant features is needed as the features are already correctly localized from the smoothing, but it is still computationally heavy.

Corners can be seen as the dominant projecting angles in a figure. A region of support for a digitized corner point consists of all the data points in the figure that bear the point as the dominant one. To Langridge (1972), each boundary point of a closed curve should have its own view of the curve and a dominant point's view constitutes a meaningful region of support of the curve, which should block the view from the neighbouring non dominant points. Teh and Chin (1989) observed that dominant points' detection depends not only on the accuracy of the measure of significance (curvatures) but mainly precise determination of support. They show that several methods give better detection when their region of support is determined well. In their paper, they propose a method that first determines the region of support for each point based on its local properties, and then computes measures of relative significance of each point. Dominant points are then detected by a process of non maxima suppression. The adaptive region of support for a point *i* is computed from the length of the chord joining points of the left and right arm and the perpendicular distance of i to the chord. The region of support is symmetric. No input parameter is needed. The algorithm seems to perform well on images consisting of multiple size features. Wu (2003) uses adaptive bending value to find region of support and a break-point detection procedure to find dominant points. Bending value for point *i* is defined as;

$$b_{ik} = max\left\{\left|\left(x_{i-k} - x_i\right) + \left(x_{i+k} - x_i\right)\right|, \left|\left(y_{i-k} - y_i\right) + \left(y_{i+k} - y_i\right)\right|\right\}$$

Starting with k=1, b_{ik} is calculated until $b_{ik} < b_{ik+1}$ and k is taken as the length of region of support of the point *i*. This method is fast and needs no input parameters.

Tsai (1997) used neural networks for a robust detection and localization regardless of object orientation in the image. Curvature here is associated with the angle between the forward and backward arms. Points with local maximum curvature are considered as corners. However the region of support still has to be identified. It gives good detection for polygonal and curved objects in arbitrary orientation but for ones with only moderate scale changes.

Mathematical morphology has also been used to detect corners. Peak and valley extraction of an object produce areas around corner points. Zhang and Zhao (1997) shrink corner portion to a single corner according to boundary information. Lin et al (1998) modified the morphological corner detectors so that they can find the actual corner points. The method is effective in detecting corners of tremendous objects but it is difficult to choose a suitable structuring element – whether a square, rhombus or circle – to fit working purpose. Liu et al (2001) uses the method on morphologic skeleton. The input source image here is represented as a polygon.

Corners are obtained by detecting the zero radius of the maximum plate on the morphologic skeleton. The method is good for noisy images, but its hardware implementations are complex as it uses a Gauss filter for noisy images. Mathematical morphology method always has one advantage over the other methods; it is fast and simple since all calculations are done only on integers, not on floating point numbers.

Many methods detect spurious corners for circular objects of varying radii. In overcoming this, Tsai et al. (1999), propose a different way of measuring corners. Their method is based on eigenvalues of the covariance matrix over a region of support. If points form a linear or almost linear line, the small eigenvalue will be close to zero. Bigger curvature between points in the region of support results in bigger values for the small eigenvalue of the covariance matrix. A point is said to be a corner if its small eigenvalue exceeds a predetermined threshold. Corners are further separated by at least *k* points, where *k* denotes the half-length of region of support. This method assumes a symmetric region of support. It gives robust detection for object shapes containing various curved and circular arcs. However the way to get the region of support was not prescribed. The authors suggested the method proposed by Teh and Chin (1989) to be used for the region of support.

Guru and Dinesh (2004) said that the left arm and right arm of a point are not necessarily of the same length and it is more reasonable and natural for the region of support to be asymmetric. They proposed a non-parametric method that used a small eigenvalue of the covariance matrix of a sequence of connected points to get an asymmetric region of support. It is known that for a straight line segment, the small eigenvalue in the continuous domain is zero. The corners are then determined based on a set of rules regarding size of region of support, limit value and curvature value. The limit value of a point *i* is the number of boundary points for which the point *i* is an end point of their region of support. Curvature at point *i* is the reciprocal of the angle made by the left and right arms at *i*. The method is said to be computationally efficient, giving no spurious corners on smooth curves and is invariant to image transformations.

Fuzzy logic has also been used to determine corner points. Lee and Bien (1996) applied fuzzy techniques to gray level images.

Sarfraz and Masood (2007d) introduced a corner detector for planar curves by sliding a set of three rectangles along the curve and counting the number of contour points lying in each rectangle. Criteria for corner detection are proposed. The method does not involve any calculation of angular measure or curvature.

3.2 Corner detection technique

Arabic fonts consist of sloping, curvy and circular arcs of various radii and curvature and as our representation of these fonts involves fitting curves to segments generated by corners, spurious corners will not only make the fonts look fragmented but will cost a lot of unnecessary calculations. It is to be noted that a visually good representation of Arabic fonts, depends much on detecting the right set of corner points.

Here curves are made up of a set of data points separated by approximately equal distance from each other. Due to the digitization process i.e. from image to data points, the data points are not a perfect construction of the curves. The collection of data points present irregularities although the curves that they represent are smooth. Thus any corner detection method that depends on angular measure (angles between points) will not be able to pinpoint accurately the right corner points and give spurious corners.

Tsai et al (1999) and Guru and Dinesh (2004) contend that using eigenvalues of covariance matrix results in less spurious corners and is most suitable for searching for corners of circular, smooth curves of various radii. However, while they use small eigenvalues of the covariance matrix, we use a different but related measure, which is better for our purpose and technique.

3.2.1 The covariance matrix

Let p_i be a data point with coordinates (x_i, y_i) and a symmetric range of support of s points. Let $S(p_i)$ be the set containing p_i and all the data points in its range of support. Thus $S(p_i)$ consists of all data points between and including p_{i-s} and p_{i+s}, that is

$$S(p_i) = \{ p_j \setminus \ j = i\text{-}s, i\text{-}s+1, i\text{-}s+2, \ldots, i, i+1, i+2, \ldots, i+s\text{-}1, i+s\}$$

Let $C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$ be the covariance matrix of $S(p_i)$. C is a 2 by 2 matrix of the covariances between the elements of $S(p_i)$, i.e.

$$c_{11} = \left[\frac{1}{2s+1} \sum_{j=i-s}^{i+s} x_j^2 \right] - c_x^2 \quad , \qquad c_{22} = \left[\frac{1}{2s+1} \sum_{j=i-s}^{i+s} y_j^2 \right] - c_y^2$$

$$c_{12} = c_{21} = \left[\frac{1}{2s+1} \sum_{j=i-s}^{i+s} x_j \, y_j \right] - c_x \, c_y \quad , \qquad c_x = \frac{1}{2s+1} \sum_{j=i-s}^{i+s} x_j$$

and $c_y = \frac{1}{2s+1} \sum_{j=i-s}^{i+s} y_j$

c_x and c_y are the geometrical centre of the curve segment $S(p_i)$. C is symmetric and positive semi-definite and thus all its eigenvalues are real and non-negative. Let λ_s and λ_l be the small and the large eigenvalue of C respectively. When $S(p_i)$ is a straight line, λ_s is zero regardless of the length and orientation of the line segment. If $S(p_i)$ is a circle, λ_s and λ_l are equal. Curve segment with sharp angles have larger λ_s than smoother ones. The smaller the radius of a circular arc, the larger λ_s will be. This is consistent with the measure of curvature of $\frac{1}{r}$ where r is the radius of the curve's osculating circle.

Figure 3.1 shows four half circles A, B, C and D, of differing radii. Their radii are measured in pixels with the smallest, having 17 and the largest, 84. The corner point of each half circle is accepted to be the middle data point. Figure 3.2a. and 3.2b. show these corner points with a support of three and six respectively. Table 3.1 and Table 3.2 show the values of the small eigenvalue λ_s , large eigenvalue λ_l and λ_r where $\lambda_r = \frac{\lambda_s}{\lambda_s + \lambda_l}$, for circular shapes A, B, C and D with support of three and six, respectively.

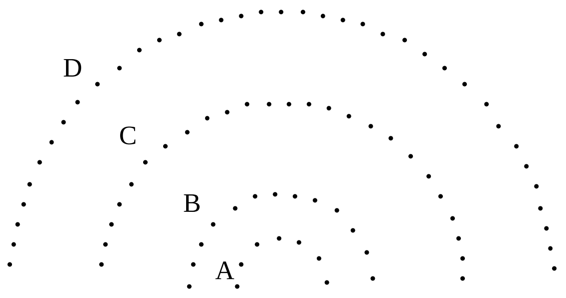

Figure 3.1 Half circles A, B, C and D with support 3, 6, 12 and 18 respectively

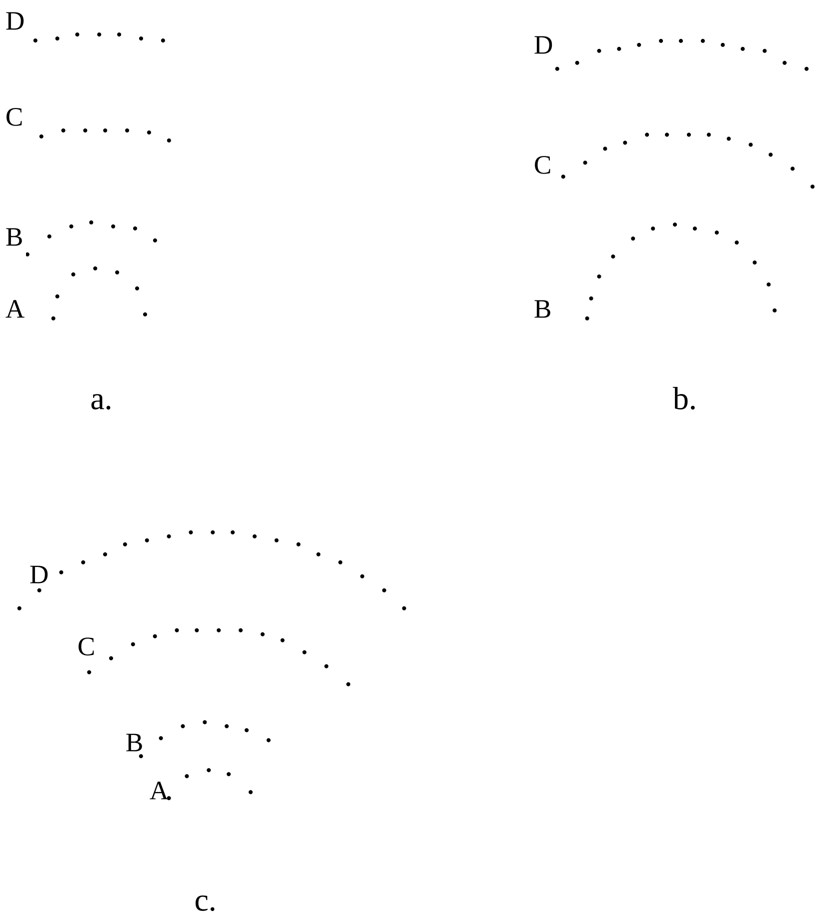

Figure 3.2 a. Circular A, B, C and D with support = 3, b. Circular B, C and D with support = 6, c. Circular A, B, C and D with support approximately halved of Figure 3.1. i.e. 2, 3, 6 and 9 respectively

Figure 3.3 shows three half ellipses A, B and C with lengths of minor axes of 71, 89 and 107, respectively. The corner point of each ellipse is the middle data