USC–SIPI REPORT #350

Geometrical Modeling and Analysis of Cortical Surfaces: An Approach to Finding Flat Maps of The Human Brain

by

Bijan Timsari

December 1999

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Geometrical Modeling and Analysis of Cortical Surfaces: An Approach to Finding Flat Maps of The Human Brain

by

Bijan Timsari

A Dissertation Presented to the Faculty of the Graduate School
University of Southern California
In Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy in Electrical Engineering

December 1999

Dedication

To my parents.

Acknowledgments

First and foremost, I wish to express my sincerest gratitude to my advisor, Professor Richard Leahy, for introducing this topic to me and for providing me the resources to pursue my research. I am grateful for the countless hours that he spent discussing with me and for his unbelievable patience in guiding me through the research. I feel extremely fortunate to have been the student of such an expert and understanding mentor.

I would also like to thank the other members of my guidance committee: Professors Ulrich Neumann, Antonio Ortega, Michael Arbib, and Alexander Sawchuk for their time and effort in evaluating and giving valuable input to my research. I also wish to thank Professors Michel Baudry and Michael Arbib who supported my research and gave me the opportunity to contribute to the USC Brain Project.

I wish to express my special thanks to the faculty and staff of SIPI for providing such an invigorating environment and for their commitment to help and support the students. They have helped me in many different ways on numerous occasions. In particular I want to thank Dr. Allen Weber for helping me with my computer related problems and Gloria Halfacre, Linda Varilla and Regina Morton for their assistance in solving my administrative problems.

During my stay at SIPI I have enjoyed the friendship of a wonderful group of people, among them I am most grateful to Stepahnie Sandor, who got me started on my research and helped me through my settlement at SIPI. I am also thankful to my other friends Chighan Hsu, James Philips, Jinyi Qi, David Shattuck, John Ermer, Sylvain Baillet and Paola Bonetto whose company made my years of graduate study an unforgettable part of my life.

Finally I want to thank my parents for the love and support they have given me in my entire life. They have made everything possible for me and I can never thank them enough.

Table of Contents

List of Figures

Abstract

In this work we have developed computational methods for creating three-dimensional models and unfolded two-dimensional isometric maps of the cerebral cortex. The three dimensional model can be used for visualization of the cortical anatomy and structural organization of the brain as well as for quantitative analysis and measurement of the geometric features of the cortical surface. The isometric flat map provides a means for effective inter-subject studies allowing visualization of the distribution of functional data over unfolded maps of the brain. The flat map is also considered as a parametric model for the cortical surface; a model appropriate for applying geometrical transformations and apt for surface based registration techniques. Although in this research the focus has been on human cortical surface, the algorithms and methods are general and equally well applicable to other objects.

Based on the fact that every conformal and equiareal mapping is isometric, we have formulated the calculation of isometric mapping between surfaces as a constrained optimization problem. We have designed an energy function whose minima occur when the surface points are positioned in an unfolded configuration. Two constraint functions imposing the requirements of preservation of angles and areas guarantee that the surface will continuously deform, subject to a simultaneous conformal and equiareal mapping. A conjugate gradient method minimizes the energy function, allowing the surface to automatically unfold and converge to a flat plane.

To perform numerical processing on the surface we have defined appropriate mathematical models to simplify the calculations as much as possible. For executing the computationally intensive iterative process of conjugate gradient we have used a simple first order approximation of the surface described as a triangular mesh. To create this model we have used our tessellation algorithm that generates meshes with fewer triangles

than other commonly used methods such as the Marching Cubes algorithm. For calculating intrinsic and extrinsic curvatures of the surface, where we need higher order approximations, we have used local parametric approximation obtained by fitting quadratic patches to the surface. We have applied these methods to computer generated phantom data and realistic physical data and obtained satisfactory results.

CHAPTER 1 *Introduction*

1.1 Complexity of The Brain

The brain is probably the most complex and the least known object in this universe. Physically, it is a sophisticated entity, an approximately1500 gram mass, containing on the order of 10^{11} neurons (the fundamental functional unit of nervous tissue), interconnected by on the order of 10^{14} synapses (functional contacts between neurons). Structurally, it is an intricate system whose circuitry and internal organization is not yet well understood. Functionally, it is an elaborate machine, the central commander of all the activities of the body with the mysterious capability of conducting the most complicated tasks. And in appearance, it is a convoluted object, covered with a highly folded 2500 cm^2 wide, 3 mm thick layer of gray matter.

1.2 Development of the Cortical Folding

The mammalian cerebral cortex is formed from six profoundly convoluted neocortical layers of gray matter surrounding a volume of fibrous white matter. The cortical foldings are thought to be the result of the gradual evolution of two initially smooth hemispheres trying to fit in a restricted volume. It is established that during the folding period the cortex continuously grows. This growth is either a uniform growth which will increase the size of the cortical surface but has no effect on its intrinsic geometry and shape, or is a differential growth that may affect both [60]. One of the earliest models proposed to explain the folding process of the cortex suggests that a cortex develops under the combined restraints

of uniform growth and minimal radial distortion and tends to fold along lines of minimal curvature [57].

In spite of the existing controversy among neuroanatomists on the relationship between the pattern of folding of the mammalian cerebral cortex and the functional and anatomical structure of the brain, there is agreement on the fact that the deviation of the surface from sphericity has a significant effect on several aspects of brain function[30]. Also, it is generally accepted that the relative expanse of cortex in more functionally complex mammals is associated with an increase in the number of functional units as opposed to an increase in the complexity of units.

There is much evidence to support the theory that the cortex is subdivided into areas, each with a specialized role. Increased specialization of cortical areas may account for the necessity for a larger brain in human beings [42]. However, it should not be assumed that all areas will be distinct and non-overlapping. Functions such as recognition, perception, motor skills and planning are divided between several areas of the cortex.

1.3 Geometry of the Cerebral Cortex

The most interesting morphological features of the cerebral cortex in mammals are the diverse and complex arrangement of their sulci and gyri. Sulci are the prominent fissures visible on the cortical surface of a mammalian brain and gyri is the name used for collective reference to the ridges that appear on the cortex. In the larger and more complicated brains, the gyri and sulci exhibit elaborate, contorted and irregular three dimensional convolution patterns. These convolutions increase the functional area of the brain without increasing the size of cerebrum beyond a biologically feasible limit.

Despite their anatomic and functional similarities, even the gyri and sulci that consistently appear in all normal anatomies exhibit pronounced variability in size and configuration.

Sulci and gyri are classified as either primary or secondary. Primary sulci are those which are most consistent in position across individuals; they tend to be deeper and appear earlier

in cortical development. Secondary sulci exhibit more variation in position and appearance [30].

The geometry of cortex can be studied from two perspectives: intrinsic and extrinsic. The extrinsic geometry of the cortex determines its anatomical appearance and the shape of white matter. The intrinsic geometry affects the relative position of functional areas and the spread of activity within the surface itself. Intrinsic geometry is more fundamental because it constrains the range of possible extrinsic properties. For the cortex, intrinsic geometry has received less attention probably because it is more difficult to measure practically or to calculate mathematically.

Previous studies of the intrinsic curvature of the cortex date back to the early 1980's when Van Essen and Maunsell reported the total amount of intrinsic curvature of cortex [60]. According to the Gauss-Bonnet theorem [22], the integral of Gaussian curvature over a surface topologically equivalent to sphere is equal to 4π. Mathematically, the two cortical hemispheres are not topologically equivalent to a sphere due to the injection of the callosum into the white matter. However, the perimeter of the hole is relatively small and the theorem applies approximately. Thus the integral of Gaussian curvature is not an appropriate measure for capturing the amount of intrinsic curvature of the cortex.

To measure the intrinsic curvature of the cortex, Griffin [30] used an alternative property called compactness defined as the ratio between the mean geodesic of the surface and the mean geodesic of a sphere of equal area. The mean geodesic of a surface is equal to the average geodesic distance between pairs of points on the surface. He showed that the cortex is compact (the compactness is less than one) and does not have the form of a large crumpled bag but in fact has an intrinsic geometry considerably different from a sphere. His experiment was a confirmation of the observations reported by Bok [5] on the human brain and Todd [58] on the ferret brain about the significant variation of the intrinsic curvature of the cortical surface. As we will see later in this thesis the Gaussian or intrinsic

curvature of the human cortex is a sharply varying quantity with large variance and mean close to zero. This implies the possibility of developing a semi-isometric mapping between the cortical surface and the plane.

1.4 Modeling the Cortical Surface

The surface of the human cortex is an incredibly complicated surface compared to the objects that are normally subject to computer modelling. Computer based surface modelling deals with objects that are either represented by a number of flat polygons or made up of smooth patches. These smooth patches are often formed by combinations of bounded parametric functions and make elegant and efficient descriptions for the surface but can be difficult to design and manipulate. Alternatively, the polygonal representation, while usually inefficient in terms of capturing the smoothness of the object, is considerably easier to deal with numerically and much easier to create in an automated way from sampled data.

Creating computer based representations of the human brain is an active area of research. In general, two class of approaches to this problem exist: contour based and voxel based, both of which use the data provided by scanning techniques such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT). Scanning devices generate a stack of slice images through an object. Each pixel in the image corresponds to some characteristic at that point in space, for example proton density in MRI or X-ray absorption in CT scans. These images together form a solid representation of the head. The contour approach attempts to trace the outline of the characteristic of interest and use these contours to form the surface. Surface reconstruction from voxel data sets typically use a variant of the Marching Cubes algorithm [37].

1.5 Mapping the Human Brain

Recent developments of computer aided medical imaging, have made it possible to record, display and manipulate 3-D models of functional activity in the living human brain, with

no apparent harm to the subject. However, the complexity and variability of cortical convolutions make it difficult to accurately and systematically analyze these functional neuroimaging data across individuals. The localization of high resolution functional activation onto highly convoluted human cortex is most easily visualized using two dimensional brain maps. By definition, 2-D maps are simplified representations of a 3-D object and their usefulness is determined by their accuracy, clarity and ability to display a particular type of information. Currently, the most accurate maps of the brain are those generated section by section from the information abstracted from 2-D samples. These are cryosection imaging based maps with very high resolution and rich information content. The major limitation of this type of map is its inability to represent 3-D information, which is a direct consequence of its two dimensional nature [4][31].

To overcome this, numerous 3-D visualization tools have been developed over the years to facilitate viewing of the brain from different perspectives and manipulating it interactively [59][43]. Three dimensional methods are necessary when we want to reconstruct a model of the brain with the size, shape and distances all preserved. While this approach can solve the problem of displaying 3-D data, it can not present the information as accurately as the cryosection maps.

A third, more abstract approach involves developing 2-D maps of the brain similar to those of the earth used for summarizing and comparing different information such as political boundaries, transportation routes and sociological data. To understand the compromises needed to generate a flat map of the brain it is useful to compare it with the simpler case of the earth. In cartography for creating a globe map, only one of the three variables, area, angle or distance, can be maintained, leading to equiareal, conformal or isometric maps. The basic approach involves placing a graticule (the network of lines of latitude and longitude) on the surface of the original object and transforming it in a systematic way onto a flat surface [54]. Over 200 different projections of the earth have

been developed, none of them ideal, and all with certain advantages and limitations depending on their intended use [55].

The surface of the brain is considerably more irregular than that of the earth, although in principal it can also be transformed to a flat plane. This dissertation is about developing a mechanism for performing such transformation.

1.6 Dissertation Goals

One of the major obstacles in performing quantitative analysis on the human brain is the difficulty of defining a simple, accurate and robust mathematical model that is suitable for the intended study. To study the functional organization of the brain it is essential to have a correct representation of the cortex. At the same time, efficiently dealing with a complex object requires the model to remain as simple as possible. The goal of this work is to develop authentic 3-D and 2-D representations of the cortical surface. To achieve this goal in particular, we have:

1. Developed an algorithm for creating a 3-D triangular approximation of the brain cortex.
2. Presented a method for calculating local differential properties of the cortical surface.
3. Defined an optimization problem for generating minimum distortion flat maps of brain.
4. Introduced a multi-resolution framework for representation of the cerebral cortex.

1.6.1 Summary of Main Contributions

A new Tessellation Algorithm: We develop an algorithm for building a triangular mesh from a set of planar contours representing coordinate-constant slices of an object. We show the advantage of this method over the Marching Cubes algorithm, an alternative method commonly used for extracting boundary surfaces from volumetric data, by comparing the results of both methods on phantom and realistic data.

Calculating the Surface Curvature: We present a method for quantifying the intrinsic and extrinsic curvatures of the surface. We calculate the minimum, maximum, mean and

Gaussian curvatures for phantom and real brain data and show that the primary folding of the cortex appear along the lines of principal curvatures.

Map Making as an Optimization Problem: We formulate the problem of making flat maps as a problem in non-convex optimization theory in which the goal is to reach a flat configuration of points with minimum distortion in area and angle and consequently in distance. We define three cost functions. The first indicates the total convexity of a surface, minimization of which results in an unfolded surface. The other two functionals measure the total difference between areas and angles of the constructing triangles of two surfaces. We evaluate the efficiency of this approach by generating flat maps for several phantoms and real data sets.

Equiareal-Conformal Map Instead of Isometric Map: We take advantage of a well known theorem in differential geometry to replace the quantitatively expensive and practically challenging process of creating isometric maps of complex surfaces with the equivalent task of generating a simultaneously equiareal and conformal map of the same surface.

Uniformly Sampled Multi-resolution Surfaces: We use the 2-D isometric flat maps to perform a uniform resampling on the original 3-D surface. Using the uniformly resampled surface we have generated a multi-resolution representation useful for multi-scale processing of the surface.

1.6.2 Overview of the Dissertation

In chapter 2, we describe our new tesselation algorithm and demonstrate its advantages over the Marching Cubes algorithm. Chapter 3 is dedicated to parametric modeling of surfaces. Here we present a method for local description of surfaces by quadratic polynomials. In chapter 4, first we provide a brief overview of some basic concepts of the

differential geometry of surfaces. Then we calculate the minimum, maximum, mean and Gaussian curvature maps of several surfaces including human cortex. Chapter 5 discusses our strategy for creating flat maps. Mathematical definitions for three of the most important types of map (equiareal, conformal and isometric) are presented followed by a complete description of the optimization framework that we use for creating these maps. Sample maps for developable as well as non-developable phantom surfaces are displayed as evidence for efficiency of our method. To prove that the objective of this research has been fulfilled, the isometric flat map of a typical cortical hemisphere is presented in this chapter. As an example of the application of isometric maps, this chapter ends with images of the uniformly resampled cortical surface in multiple resolutions. Finally, chapter 6 describes possible extensions of this work.

CHAPTER 2 *Tesselation of Human Cortical Surface*

Development of a mathematical representation of the brain surface suitable for automatic feature identification is often a prerequisite for brain visualization, quantitative analysis of brain properties, multimodal registration, and mapping and unfolding the cortical surface. With the current state of brain imaging technology, MR imaging is the only noninvasive technique that generates the anatomical details of the brain with a resolution high enough for reasonable quantitative measurements of geometric properties of the brain. Extracting the human cortical surface from MR images has been a topic of interest for many researchers over the past two decades. A range of computerized techniques has been proposed for this purpose with varying level of automation. Difficulties such as image intensity inhomogeneities, extremely convoluted cortical structures, and topology pose difficult challenges. Many of these techniques use an intensity based approach, and attempt to determine appropriate thresholds for an initial segmentation. Some methods are semi-automated and their role is only to provide an initial segmentation to assist a trained expert to do the final labeling of brain structures [64]. Fully automated methods have also been proposed. Some of them employ statistical image analysis techniques [62][34], others use morphological image processing [8][6][45]. There are still other methods that take an indirect approach and register the brain to a reference atlas and then perform the labeling through tissue matching with respect to the atlas [14].

2.1 Survey of Current Techniques

The immediate output of a brain surface extraction algorithm is an unorganized set of points defining either the outer or inner boundary of cortical gray matter. The problem of reconstructing a three-dimensional surface from a set of points is a major practical obstacle in such diverse fields as medical imaging and computer aided geometric design. The problems are not limited to finding the best interpolants, but also concern the definition of the shape of the object, estimates of geometric properties such as area, volume, curvature, and extraction of elementary shapes. The difficulty of the problem increases with the complexity of the shape of the object. As with any other practical application, computational resources are always a limiting factor in computational complexity.

Solutions to this problem fall into one of the two broad categories of volume based and surface based methods. Volume based approaches assume that the data are available as a set of discrete values defined on a 3-D grid. Several methods have been proposed for surface reconstruction based on such an arrangement of data, the most popular of which is the Marching Cubes algorithm [37] and its extensions. The Marching Cubes technique compares the intensity values at eight contiguous voxels from two adjacent slices forming a cube with a predefined isovalue to determine how the surface intersects this cube. Trilinear interpolation along each of the voxel edges is used to find the vertices of the triangular patch that forms part of the isosurface passing through a given voxel. The Marching Cubes algorithm can be applied directly to the original brain MRI; however, due to the limited efficiency of the threshold based schema it uses for determining the location of surface points, it is often preferable to generate the brain surface data by using a custom designed segmentation method. Marching Cubes is then applied to the resulting binary level image defining the desired brain volume.

One obvious problem with Marching Cubes is the large number of vertices, and consequently triangles, needed to represent the surface. As each boundary cube can generate up to 4 sub-pixel facets, the resulting set of triangles can be quite large. The original algorithm proposed in [37] has several ambiguous cases which can lead to holes in the final image [24]. These ambiguities occur either when the wrong decision is made about whether a vertex is inside or outside a surface, or if the triangles chosen do not fit together properly. A number of extensions have been proposed to the Marching Cubes algorithm to avoid the ambiguous cases. These include marching tetrahedra [49] and dividing cubes [13].

The input data to a surface based approach are a series of planar contours corresponding to the intersections of the object surface with a set of parallel planes. A polyhedral approximation of the surface is obtained by linking each set of contours from two consecutive slices with triangular patches. The desired surface is the best approximating polyhedron. The difference between the methods in this category lies in the definition of the best approximation. Different metric functions have been used to distinguish good surfaces from bad [50][51]. Some methods optimize a cost function based on a graph describing the set of all possible arrangements of triangular patches to determine the desired surface. In this formalism a given path in the graph defines a possible solution. The optimal path may be found by maximizing the volume [35] or minimizing the area [28] of the polyhedral representation of the surface, or by optimizing some global figure of merit assigned to individual triangles through a relaxation process [61]. A variety of other metrics for the graph cost function has been used, each claiming to improve some "good" feature of the surface [40][51]. These optimal methods are generally slow and time consuming. Other methods take advantage of less computationally expensive heuristic methods to create visually acceptable surfaces. They define triangular patches one by one using a local decision criteria [12][15][16][29]. Heuristic methods can generate incorrect

results if the contours are very different in shape, orientation and alignment. Furthermore, most of these algorithms fail to handle ambiguous cases that result from having multiple contours in a single section (correspondence ambiguity) or different numbers of multiple contours in adjacent sections (branching ambiguity).

To solve the correspondence problem some methods make the assumption that the data is sampled densely enough that by examining the overlap between contours on adjacent sections, unambiguous solutions are found. More general methods incorporate high level information about the shape of the objects in the reconstruction process [52][7]. Proposed solutions for the branching problem include algorithms that require user interaction [12][65] as well as methods that compute Delaunay triangulation of the set of vertices belonging to the contours in adjacent sections [3]. A problem related to the branching problem is tiling a pair of contours with very different shapes. The proposed solution to this problem consists of creating an interpolated contour between the two adjacent slices, and thus reducing the dissimilarity between contours to be tiled [44][25].

2.2 Triangulation Algorithm Based on Coordinate Curves

The common strategy in surface based triangulation algorithms is to reconstruct the external surface of the object using a set of contour points extracted from one series of its parallel planar sections. The polyhedral approximation of the surface is obtained by linking contour points in consecutive sections with triangular patches. These algorithms were originally developed to conform to the structure of the data acquired by CT and MRI techniques in which a three-dimensional image of a body's structure is constructed by computer from a series of plane cross-sectional images made along an axis. Representing a three dimensional object by a series of two dimensional sections hides the information about the connectivity of sections along the imaging axis; however, it provides sufficient data for finding points located on the surface of the object and their neighborhood relations in the plane of imaging. Since the volume image data acquisition is based on a

rectangular grid, one can always rearrange the data set to effectively change the imaging axis and find the neighborhood relations in all three sets of mutually orthogonal planes defining the imaging grid.

Mathematically, a point located on a surface with a parametric description of $S(u,v)$ is uniquely specified by two constant parameter (coordinate) curves passing through it. In a rectangular coordinate system, the contours extracted from the cross sections of an object with planes normal to any of the three axes correspond to coordinate curves. Thus each surface point is defined as the point of intersection between contours belonging to two perpendicular planes. We have developed a two step algorithm for triangulating the surface that is based on this property.

2.2.1 Contour Tracing

In a cross-sectional image of a three dimensional object, boundaries of regions correspond to intersections of the object's surface with the plane of sectioning. The connectivity structure of the surface is directly related to how the boundary points are defined. If an 8-neighborhood system is used to define boundary points, the resulting contours will be 4-connected, and correspondingly, boundary points defined based on a 4-neighborhood system result in 8-connected contours (Figure (2.1)).

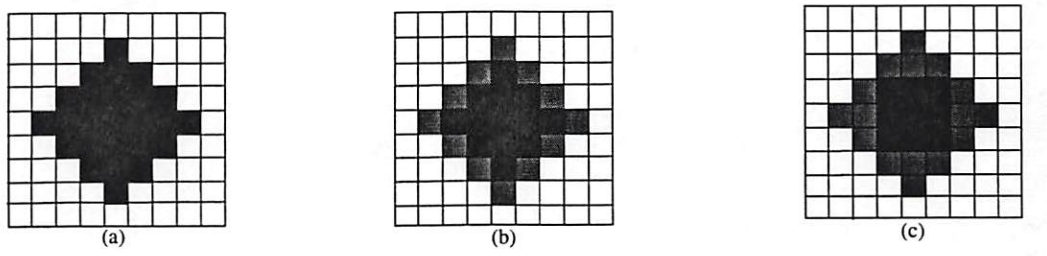


Figure 2.1. 8-Connectivity vs. 4-Connectivity. (a) Sample shape. (b) Boundary points extracted based on 4-neighborhood system are 8-connected. (c) Boundary points extracted based on 8-neighborhood system are 4-connected.

The union of the three sets of 8-connected contour points extracted from sections perpendicular to coordinate axes constitutes an 18-connected surface. Figure (2.2) displays the eighteen closest neighbors of a point located in the center of the $3 \times 3 \times 3$ cube.

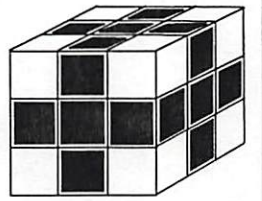


Figure 2.2. 18 closest neighbors of a point.

By tracing the contours in a consistent manner (either in clockwise or counter clockwise direction) the two closest neighbors of each surface point in the plane of the contours are determined. Where a point is located in the intersection of multiple contours in a single plane, more than one pair of closest neighbors are found for that point. This is the situation where two constituting elements of the object are tangent to each other. By separating the neighbors into two pairs we guarantee that in the tiling step the correct surface representation of each element will be constructed.

A typical surface point has three pairs of closest neighbors lying on three mutually orthogonal planes that uniquely localize the point in space. A triangle formed by connecting the point to two of these six neighbors is the smallest possible patch, within the limits of the image resolution, that one can use to tile the surface. The surface constructed from such triangles will have a resolution corresponding to that of the original image.

2.2.2 Tiling

Surface points can be divided into two groups depending on the number of contours passing through that point. Those that are in the intersection of contours lying on two perpendicular planes and those that are in the intersection of contours belonging to three mutually orthogonal planes.

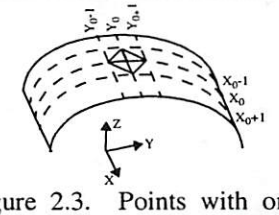


Figure 2.3. Points with only two pairs of neighbors are tiled with four equal right triangles.

Points in the first group are located on planar regions to which one of the coordinate axes is locally orthogonal. This group of points have only two pairs of closest neighbors to

which they can be linked with four equal right triangles. As an example, Figure (2.3) shows the point (X_0,Y_0Z_0) that is shared between a constant-X and a constant-Y curve and is located on a patch that is contained in the $(Z = Z_0)$ plane. The four nearest neighbors of this point are $(X_0 \pm 1,Y_0, Z_0)$ and $(X_0, Y_0 \pm 1, Z_0)$.

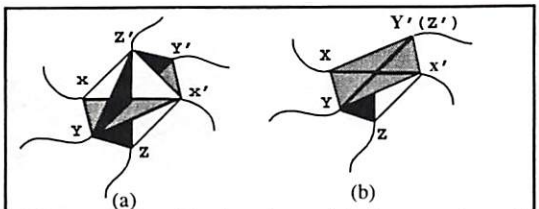


Figure 2.4. (a) A point with three pairs of neighbors forms 12 triangles. (b) A point with overlapping neighbors forms less than 12 triangles, neighbor points Y' and Z' are overlapped and only 8 triangles are formed.

The second group consists of points that have at least three pairs of closest neighbors on the surface. A point of this type can form up to twelve triangles when linked to all its neighbors (Figure (2.4.a)). In some locations one of the neighbors in one pair may coincide with a neighbor in another pair. Such an arrangement of points will result in less than twelve triangles (Figure (2.4.b)).

One can not simply join all the points to their neighbors with triangular patches to tile the surface since this would result in many redundant triangles and doubly covered regions. Instead, by orderly tiling different regions of the surface with triangles selected based on some constraints, a relatively uniform triangulation of the surface can be obtained.

The unit tiling patch is composed of two triangles with a common side (Figure (2.5)).When a point is to be connected to two of its neighbors, this unit patch is fit to the surface by superimposing the common side of the triangles on the edge linking the neighbors of that point. To avoid double coverage of the surface, after fitting a unit patch to a set of four points no other patch will be allowed to cover that set.

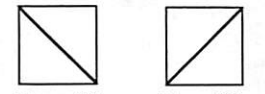


Figure 2.5. Two types of unit tiling patch.

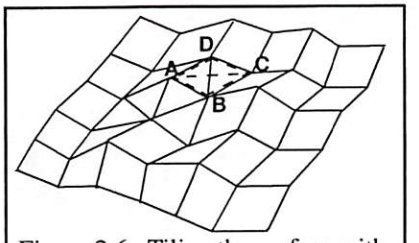


Figure 2.6. Tiling the surface with a unit patch at a typical point.

In Figure (2.6) we have shown a small region of a surface to explain how the tiling algorithm works at a typical point A. First we pick two neighbors of A, namely B and D. We then find their other common neighbor, C. We cover the patch limited to these four point with triangles ABD and CBD and label this patch as processed. When we move to point B, neighbors A and C will be ignored since they have already been processed. The same rule applies to point C with neighbors B and D and point D with neighbors A and C. Depending on the order in which the points are being processed, the shape of the triangulation may change. For example if point B was processed before point A then ABCD would be tiled with a unit patch of type(2) instead of type(1) (dotted triangles in Figure (2.6)).

However, this does not have any effect on the shape of the constructed surface in planar areas where all the four points are coplanar. In other areas the resulting change in the shape of the surface is on the same scale as the voxel size and either result could be considered "correct".

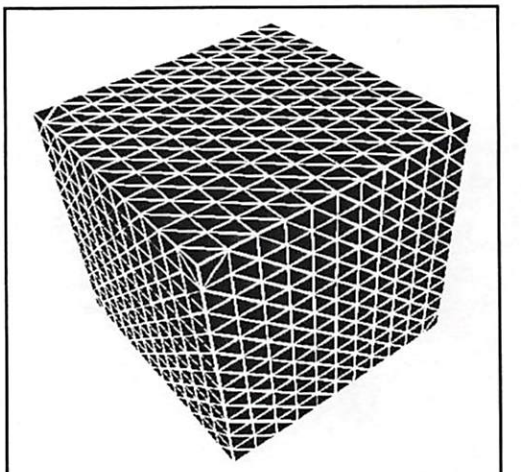


Figure 2.7. Tessellation of a cube using coordinate curves.

As we described in Section (2.2.1), surface points located on planar patches parallel to coordinate planes (normal to coordinate axes) have only two pairs of equidistant neighbors. These points, when tiled with unit patches, create a surface made of regularly arranged equal right triangles. Figure (2.7) shows a regularly triangulated cubic surface constructed with this algorithm.

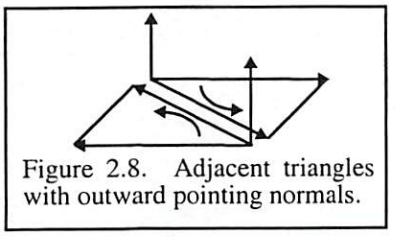


Figure 2.8. Adjacent triangles with outward pointing normals.

Once the entire surface is tiled all of the triangles need to be checked for orientation consistency. The vector normal to the plane of a triangle represents the orientation of the surface at that location and it must point outward with respect to the object. Since our tiling operation is a local process and the outward direction is determined using global information, for each point on the object we have to calculate and store the outward pointing vector before beginning the tessellation. Alternatively, we can select the simpler option of reorienting wrongly oriented triangles. The normal vector to the plane of a triangle is calculated by taking the cross product of the vectors formed from two sides of the triangle, therefore the normals to two adjacent triangles sharing a side will both point in the same direction if vectors representing the common side are in opposite directions (Figure (2.8)). Starting from a triangle whose normal is pointing outward, we find all adjacent triangles and if necessary change their orientations so that every pair of adjacent triangles have consistent orientations as described.

2.3 Results and Discussion

To evaluate this method we compared its performance with the Marching Cubes algorithm [37]. To compare the ability of the two methods to reconstruct surfaces with sharp corners vs. surfaces with smooth curves we used a phantom made of a hemisphere attached to a cube. Figure (2.9.a) shows the surface generated by our method consisting of 2894 vertices and 5784 triangles. Figure (2.9.b) shows the output of the Marching Cubes algorithm. This surface is made up of 3782 vertices and 7560 triangles. In Figure (2.9.c) we have resampled the object to make the number of surface points used in our method close to the number of vertices generated by the Marching Cubes algorithm. The resampled object resulted in 4130 vertices and 8256 triangles. The Marching Cubes method rounds right-angled edges whereas our method preserves them. However, when constructing smooth surfaces our method compared to Marching Cubes creates a rougher

surface. This is due to the smaller number of vertices and triangles. By increasing the number of vertices we observe that with a comparable number of triangles, our methods creates a smoother surface.

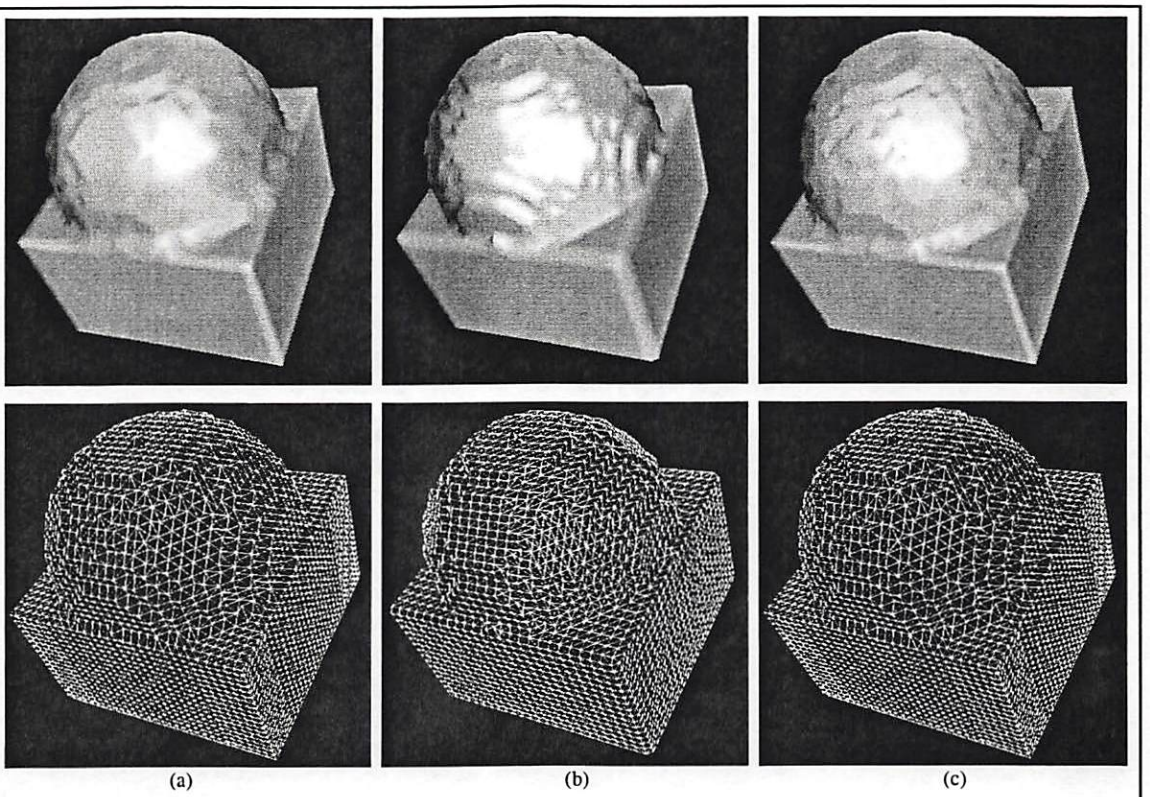


Figure 2.9. (a) Phantom surface constructed from coordinate curves (2894 vertices, 5784 triangles). (b) Phantom surface tessellated using Marching Cubes algorithm (3782 vertices,7560 triangles). (c) Resampled phantom constructed from coordinate curves (4130 vertices, 8256 triangles).

The fact that we use coordinate curves to determine the location of surface points implies that we should expect to see a staircase shape for the reconstructed surface of objects with planar faces that are not positioned parallel to any of the coordinate axes. To examine if the same effect will be observed when the object's surface is reconstructed by the Marching Cubes method we applied both methods to a rotated version of the phantom used in the previous study. The results are displayed in Figure (2.10) and Figure (2.11). It can be seen that both methods suffer from the staircase effect.

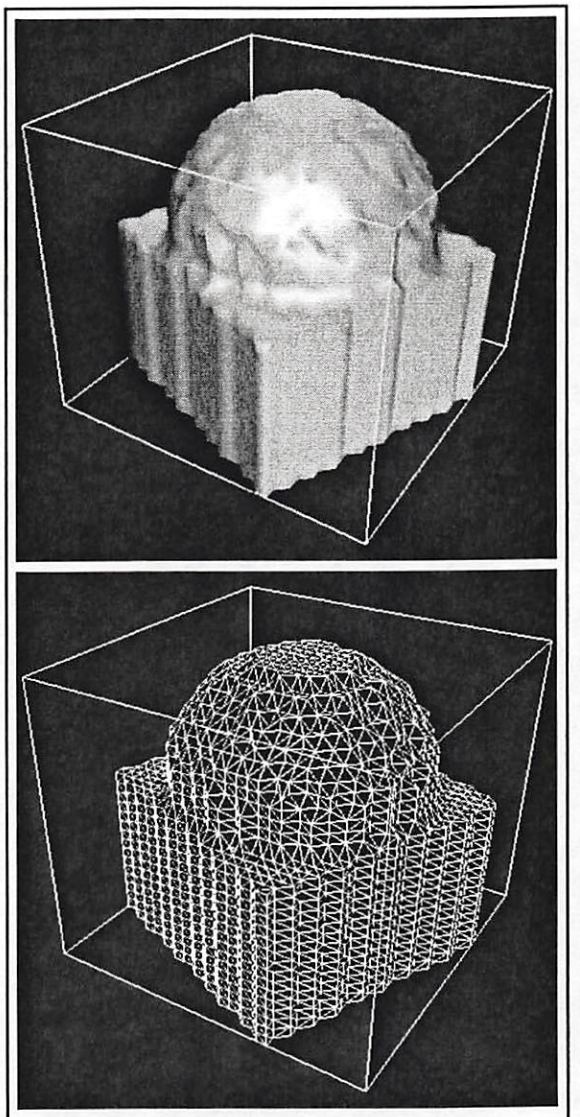


Figure 2.10. Reconstructed surface of the rotated phantom using the coordinate curves method.

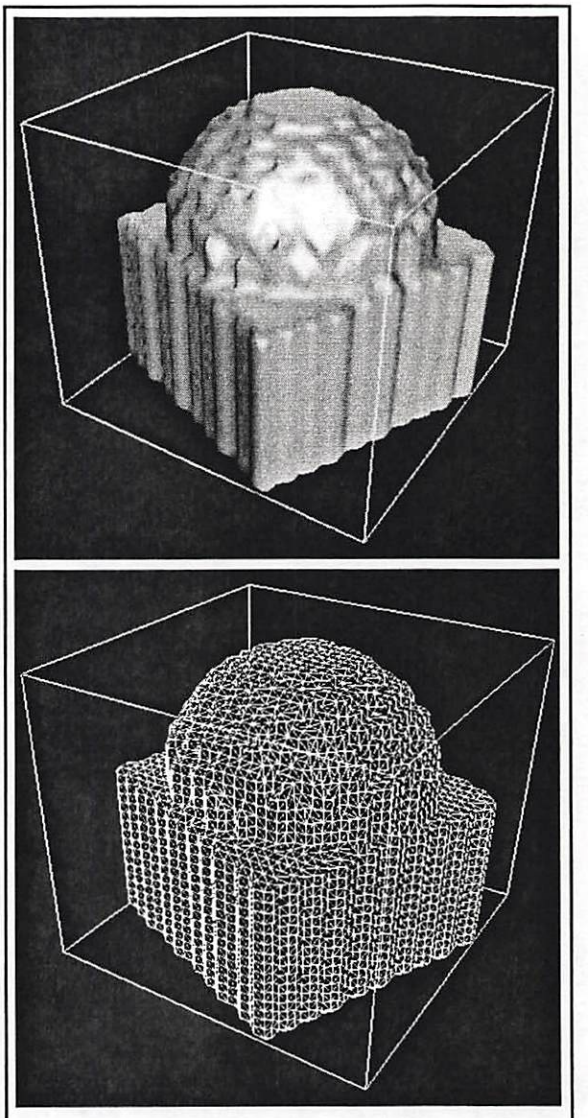


Figure 2.11. Reconstructed surface of the rotated phantom using the Marching Cubes algorithm.

To compare the outputs of these methods for real medical images we used a data set representing the cortical surface of a human brain extracted from 150 axial magnetic resonance images of the head of a normal subject, acquired with a total of 256×256 voxels per slice. After tracing the contours in transaxial, coronal and sagittal sections, a total of 73814 points were found to represent the outer surface of the brain. These surface points were then linked with 147645 triangles to create the surface displayed in Figure

(2.12.a). For the same object, the Marching Cubes algorithm generated 128984 vertices and 258004 triangles (Figure (2.13.b)). Using the same parameter settings for rendering the two surfaces one can not observe any major degradation in the visual quality of the surface generated by our method. Given the significant decrease in the number of vertices and triangles without substantially compromising the quality of the resulting surface this method is a potentially attractive alternative to the Marching Cubes algorithm.

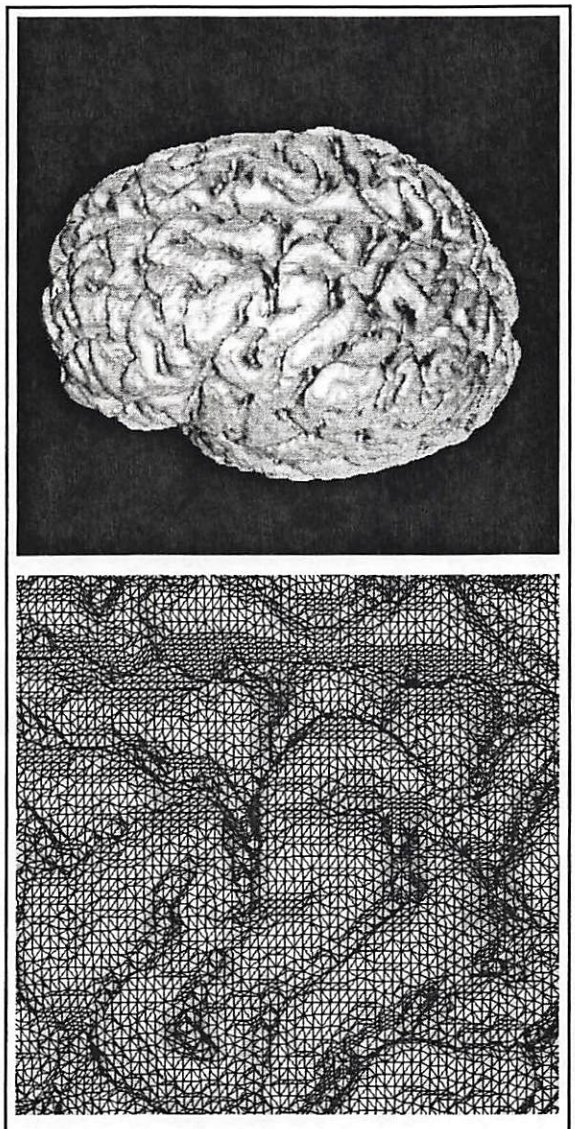


Figure 2.12. Human brain surface reconstructed using coordinate curves. No. of vertices: 73814 No. of triangles: 147645.

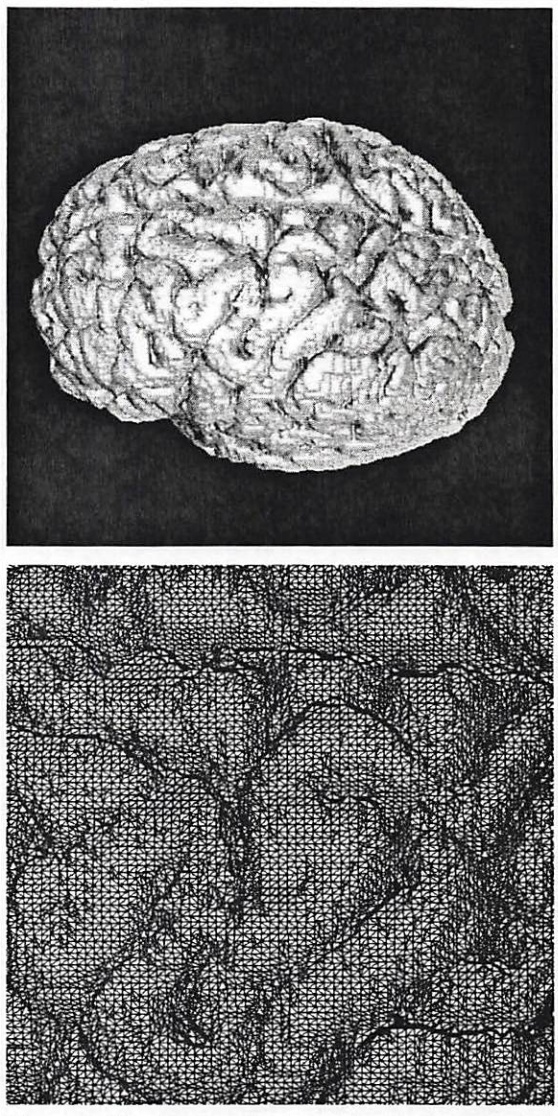


Figure 2.13. Human brain surface reconstructed using Marching Cubes. No. of vertices: 128984 No. of triangles: 258004.

CHAPTER 3 *Parametric Modeling of Human Cortical Surface*

Differential geometry is the mathematics of using calculus to study geometric properties of curves and surfaces. Although biological surfaces (and in particular the cortex) have a non-zero thickness and so are not true surfaces, considering the relatively constant small thickness (3-4 mm) of human cerebral cortex compared to its large surface area (2000-2500 cm^2), application of the techniques of differential geometry in the analysis of human cortical surface can be shown to be useful [30][57]. In order to appreciate geometric properties of the brain and its effects on cortical architecture and function we need to develop a precise mathematical representation adaptable for differential analysis of the cortex. This representation allows us to make statements about the geometry of the surface and study the variability of the cortical surface in an ensemble of brain data.

3.1 Survey of Current Techniques

The problem is to fit a mathematical model to a surface given as a piecewise linear function representing a collection of triangular faces pasted together along their edges. For a detailed description of the available methods for solving this problem refer to [41]. Here we present only a brief overview of the general approaches to this problem.

Two general types of surface fitting algorithms exist, interpolation and approximation. In interpolation the constructed surface satisfies the given data precisely and the number of surface pieces is approximately the same as the number of data points. A triangular mesh

is the lowest order interpolating model for a surface. As the number of data points increases the space and computational requirements of interpolated shape representations become a major disadvantage. In approximation, we construct surfaces that do not necessarily satisfy the given data precisely, but only approximately. In this case it is often desirable to specify a maximum bound on the deviation of the surface from the given data, and to specify some constraints, i.e. data that is to be satisfied precisely.

Most fitting algorithms also fall into one of two categories: global or local. In a global algorithm, a system of equations or an optimization problem defined collectively in terms of all the data points needs to be solved. Theoretically, a perturbation of any one input data item can change the shape of the entire surface; however, the magnitude of the change decreases with increasing distance from the affected data item. Local algorithms are more geometric in nature, constructing the surface segment-wise, using only local data for each segment. A perturbation in a data item only changes the surface locally. These algorithms are usually computationally less expensive than global methods.

3.2 Local Fitting of Quadratic Patches to a Triangular Mesh

To study the differential geometry of human cerebral cortex we need to measure its local properties. That means those properties that depend only on the behavior of the surface in the neighborhood of a point. The mathematical representation resulting from a local surface fitting method is adequate for studying these properties. To use methods of differential calculus, the function describing the surface must satisfy continuity and differentiability requirements appropriate to the property under study. Our goal is to measure different curvatures of the human cortical surface including minimum, maximum, mean and Gaussian curvature. A model that can present meaningful values for the curvatures of human cortex must at least belong to the class of smooth regular surfaces of order two (C^2), i.e the surfaces defined by the mapping $X(u, v) : U \rightarrow R^3$ from an open set $U \subset R^2$ into R^3 such that all the partial derivatives of order two or less are continuous in

U, and $X_u \times X_v \neq [0, 0, 0]^t$ for all $(u, v) \in U$, where X_u and X_v are partial derivatives with respect to u and v and x is the cross product [22].

To keep our model as simple as possible we have chosen a model satisfying the minimum requirements based on the schema described by Hamann [32]. At each vertex in the triangular mesh, a quadratic function is fit by locally approximating the parametric surface by using terms up to order two in its Taylor series.

We assume that a triangulated mesh is constructed from the contours representing the surface using the algorithm described in Chapter 2. Let $X_i \in R^3$, $i = 0, 1, \ldots, N-1$ represent the vertices of the mesh and $\mathcal{N}^K(X_i)$ the set of all Kth-order neighbors of vertex X_i. We define the Kth-order neighbors of a point X_i as all the $1st$-order neighbors of its $(k-1)th$-order neighbors that are not common with its $(k-2)th$-order neighbors, with the $1st$-order neighbors defined as the immediate neighbors of X_i and each point defined as its own $0th$-order neighbor.

At each vertex, X_i, we calculate the surface normal, n_i, by averaging the normals to the planes of triangles connected to that vertex, i.e.

$$n_i = \frac{1}{N_i} \sum_{j=0}^{N_i - 1} \frac{a_j \times b_j}{\|a_j \times b_j\|} \qquad \text{(Eq. 3.1)}$$

where a_j and b_j are the vectors representing the sides of the triangles connected to vertex X_i and N_i is the total number of such triangles. The order of the vectors is selected such that the cross product always points outward with respect to the surface (Figure (3.1)).

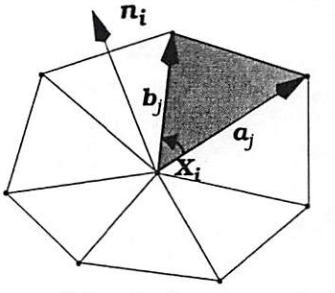


Figure 3.1. Surface normal at a typical point of a triangulated mesh.

The tangent plane T_P at a point X_i of the surface is defined by $n_i \cdot (X - X_i) = 0$ where n_i is the unit surface normal at X_i. We define an orthonormal coordinate system in T_P as follows:

First, based on the outward unit normal vector n_i, we define a vector m_i perpendicular to n_i according to

$$m_i = \begin{cases} \frac{1}{n_{ix}}[-(n_{iy}+n_{iz}), n_{ix}, n_{ix}]^t, & if\ n_{ix} \neq 0 \\ \frac{1}{n_{iy}}[n_{iy}, -(n_{ix}+n_{iz}), n_{iy}]^t, & if\ n_{iy} \neq 0 \\ \frac{1}{n_{iz}}[n_{iz}, n_{iz}, -(n_{ix}+n_{iy})]^t, & if\ n_{iz} \neq 0 \end{cases} \quad \text{(Eq. 3.2)}$$

and then form the coordinate system using three orthonormal vectors (figure(3.2)):

$$\left(n_i\ , \quad P_i = \frac{m_i}{\|m_i\|}, \quad Q_i = n_i \times P_i \right) \quad \text{(Eq. 3.3)}$$

The quadratic patch approximating the parametric surface passing through the point X_i can be written as:

$$X(u,v) = X_i + uP_i + vQ_i + ([u,v]C_i[u,v]^t)n_i \quad \text{(Eq. 3.4)}$$

where C_i is a 2×2 symmetric matrix. The unknown matrix C_i is determined by solving a minimization problem defined as:

$$C_i = \arg\left(\min_{C} \sum_{X_j \in \mathcal{N}^K(X_i)} \left\| X_j - \left\{ X_i + u_j P_i + v_j Q_i + ([u_j, v_j]C[u_j, v_j]^t)n_i \right\} \right\|^2 \right) \quad \text{(Eq. 3.5)}$$

where $X_j \in \mathcal{N}^K(X_i)$ represents all the Kth-order neighbor points of vertex X_i, with $K \geq 1$. Let $X_j^P = [u_j, v_j]^t$ be the projection of point X_j into the tangent plane T_P. The local coordinates of the points X_j^P in the tangent plane are determined by:

$$[u_j, v_j]^t = [(X_j^P - X_i)\cdot P_i, (X_j^P - X_i)\cdot Q_i]^t \quad \text{(Eq. 3.6)}$$

We can restate the minimization problem in terms of distances of the points in the neighborhood $\mathcal{N}^K(X_i)$ from the tangent plane. We denote such distances with h_j, i.e. $h_j = (X_j - X_j^P)\cdot n_i$. Since $X_j^P = X_i + u_j P_i + v_j Q_i$, we can write

$$C_i = \arg\left(\min_{C} \sum_{j} \left| h_j - ([u_j, v_j] C [u_j, v_j]^t) \right|^2 \right) \qquad \text{(Eq. 3.7)}$$

By introducing matrices:

$$U_i = \begin{bmatrix} u_0^2 & 2u_0v_0 & v_0^2 \\ \cdots & \cdots & \cdots \\ u_{N_{i-1}}^2 & 2u_{N_{i-1}}v_{N_{i-1}} & v_{N_{i-1}}^2 \end{bmatrix}, \; C_i = \begin{bmatrix} C_{11} \\ C_{12} \\ C_{22} \end{bmatrix}, \text{ and } H_i = \begin{bmatrix} h_0 \\ \cdots \\ h_{N_{i-1}} \end{bmatrix}$$

we can use matrix algebra and rewrite the problem as

$$\hat{C}_i = \arg\left(\min_{C} \left\| U_i C_i - H_i \right\|^2 \right) \qquad \text{(Eq. 3.8)}$$

The minimum mean squared estimate of C_i can be easily found by solving the following normal equations [53]:

$$U_i^t U_i \hat{C}_i = U_i^t H_i \qquad \text{(Eq. 3.9)}$$

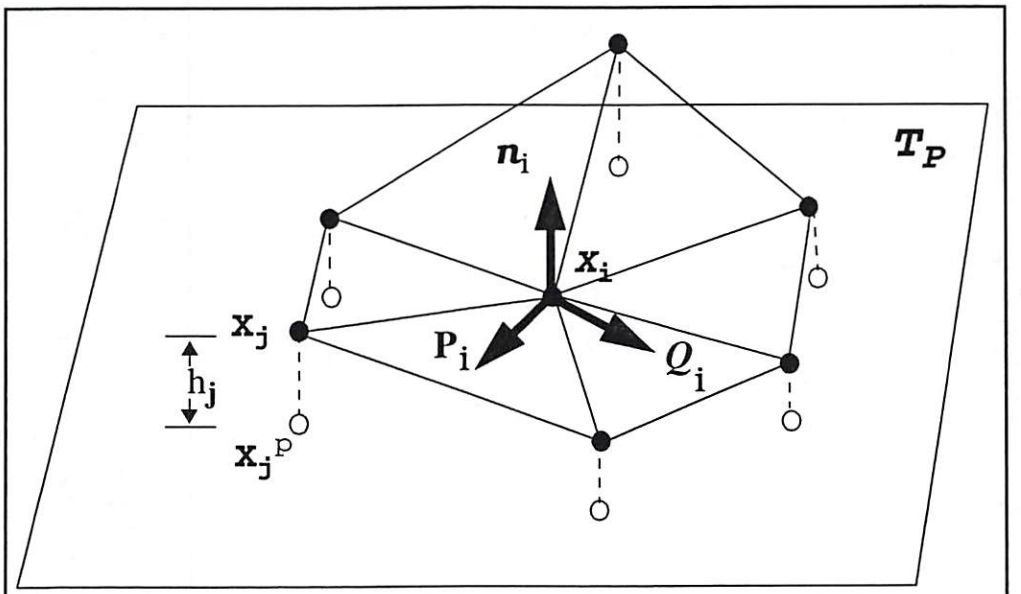


Figure 3.2. The local rectangular coordinate system in the neighborhood of a point $\mathbf{x_i}$.

To study the accuracy of the surface description obtained based on this method we used the spherical phantom displayed in Figure (3.3). We selected a small region of the phantom (Figure (3.3.b)) and fit a quadratic patch to it. The center of patch is marked with a black dot in Figure (3.3.b). As it can be seen in Figure (3.3.c) and (d) the accuracy of the quadratic patch in describing the surface decreases with distance from the center of the patch.

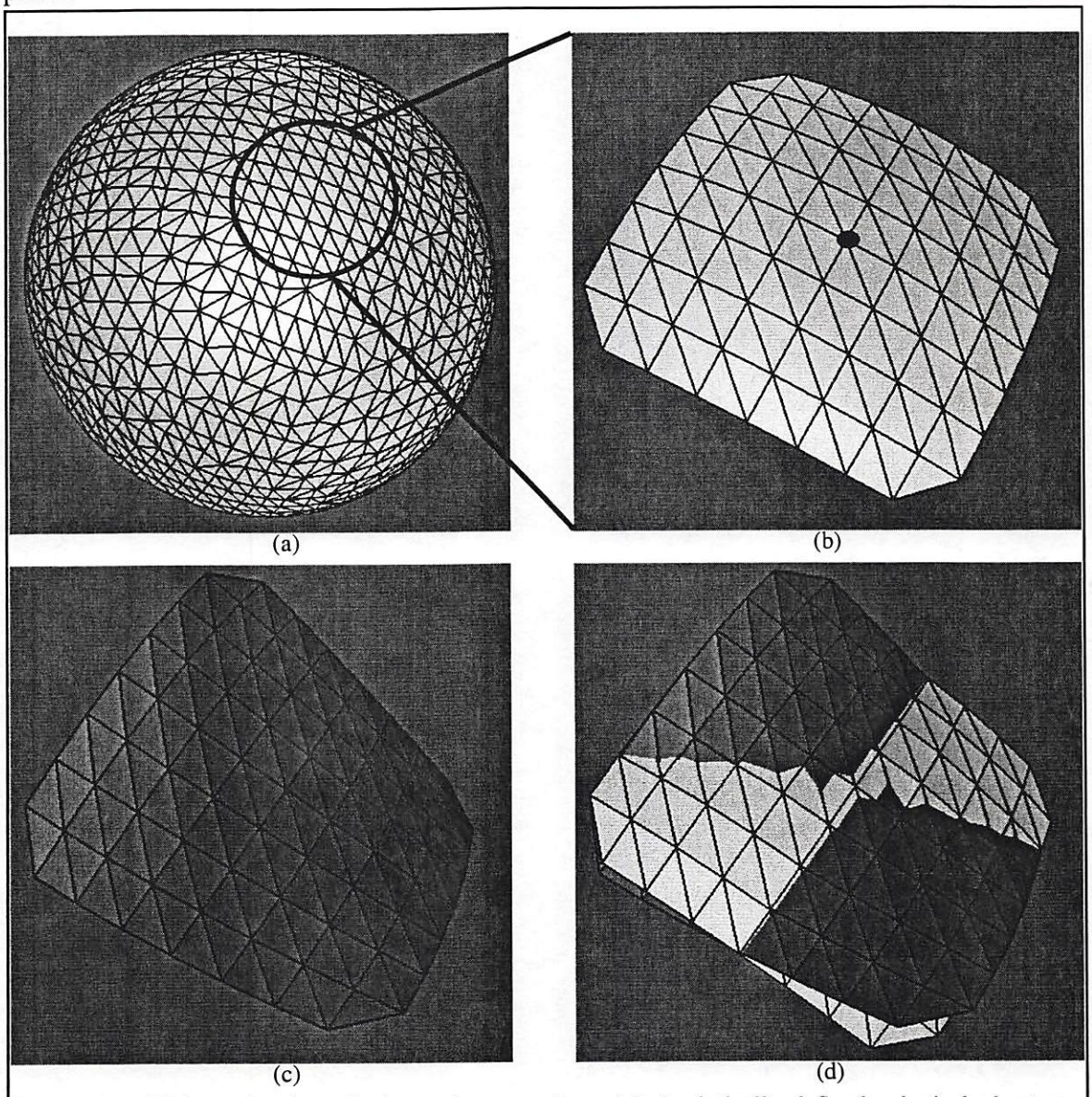


Figure 3.3. Fitting a local quadratic patch to a sphere. (a) Analytically defined spherical phantom. (b) Magnified display of a small region of sphere. (c) The same region approximated with a local quadratic patch. (d) Superimposition of the original and approximated patches.

CHAPTER 4

The Curvature of Human Cortical Surface

4.1 Introduction

In this chapter we want to study local geometric properties of the cortex. In other words, our focus will be on the properties that do not pertain to the geometric configuration of the surface as a whole but depend only on the form of the configuration in an arbitrary small neighborhood of a point under consideration. Among many local geometric quantities defined for the surface, curvature information is the most valuable since it quantifies the structure of the sulci and gyri, thus providing the basis for comparative studies.

Curvature is a local property indicating a measure for how rapidly a surface pulls away from the tangent plane in a neighborhood of a point. In what follows we will introduce the relevant concepts (principal curvatures, principal directions, Gaussian and mean curvatures) without using local coordinates to emphasize the geometric content of the definitions. However, for computational purposes, it is important to express all concepts in local coordinates.

Throughout our discussion we assume that $\boldsymbol{X}(u, v) = (x(u, v), y(u, v), z(u, v)) : U \rightarrow R^3$ is the parametric representation of the surface defined on U, an open subset of R^2 with coordinates u and v and it satisfies all the required conditions. We therefore make the following assumptions:

(1). X is differentiable. This means that the surface $X(u, v)$ belongs to the class of smooth surfaces of order r, C^r $(r \geq 1)$, i.e. the functions $x(u, v)$, $y(u, v)$, $z(u, v)$ are r times continuously differentiable in U.

(2). X is a homeomorphism. Since X is continuous this means that X has an inverse X^{-1} which is continuous.

(3). X is regular. This means that for each $P \in U$, the differential $dX_P : R^2 \rightarrow R^3$ is one-to-one or equivalently the Jacobian matrix

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix} \qquad \text{(Eq. 4.1)}$$

is of rank 2 in U. Still another way of expressing this condition is that $\frac{\partial X}{\partial u} \times \frac{\partial X}{\partial v} \neq 0$, which means there is no singular point on the surface i.e. a unique unit surface normal always exists at each point of the surface.

4.2 Some Basic Concepts of Differential Geometry of Surfaces

There are metrics for a surface that can be calculated without leaving the surface. Geometrically, these metrics describe quantities such as lengths of curves, angles of tangent vectors and areas of regions that can be measured on the surface without referring to the ambient space R^3 where the surface lies. Such properties are based only on the *first fundamental form* of the surface.

DEFINITION 1. The infinitesimal distance element between two points on a curved surface is given by $ds^2 = X_u \cdot X_v du^2 + 2X_u \cdot X_v du dv + X_u \cdot X_v dv^2$ and is called the *first fundamental form* of the surface. Furthermore, the following quantities are called the *first fundamental* or *metric coefficients* of the surface.

$$E_X(u, v) = X_u \cdot X_u \quad \text{(Eq. 4.2)}$$

$$F_X(u, v) = X_u \cdot X_v \quad \text{(Eq. 4.3)}$$

$$G_X(u, v) = X_v \cdot X_v \quad \text{(Eq. 4.4)}$$

The first fundamental form gives the distance ds between neighboring points (u, v) and $(u + du, v + dv)$ on a surface, to first order in du and dv. The distance element ds lies in the tangent plane of the surface at (u, v) and therefore yields no information on how the surface curves away from the tangent plane at that point. To investigate surface curvature, we must examine the displacement between neighboring points (u, v) and $(u + du, v + dv)$ to second order in du and dv. This is what can be determined from the *second fundamental form* of a surface.

DEFINITION 2. The projection of the second order infinitesimal displacement between two points of a curved surface on the unit surface normal is equal to one half of the second fundamental form of the surface given by $d^2h = \boldsymbol{n} \cdot X_{uu}du^2 + 2\boldsymbol{n} \cdot X_{uv}dudv + \boldsymbol{n} \cdot X_{vv}dv^2$, in which $\boldsymbol{n}$ represents the unit surface normal. The following quantities are called the second fundamental coefficients of the surface and form the basis for defining and analyzing the curvature of a surface.

$$L_X(u, v) = \boldsymbol{n} \cdot X_{uu} \quad \text{(Eq. 4.5)}$$

$$M_X(u, v) = \boldsymbol{n} \cdot X_{uv} \quad \text{(Eq. 4.6)}$$

$$N_X(u, v) = \boldsymbol{n} \cdot X_{vv} \quad \text{(Eq. 4.7)}$$

We are going to study the geometric shape of a surface X in the neighborhood of a typical point $\boldsymbol{P}$. We consider all surface curves $C:(u = u(t), v = v(t))$ passing through $\boldsymbol{P}$ and denote their curvatures by $\kappa = \kappa(t)$. We use symbol $\gamma = \gamma(t)$ to represent the angle between the unit principal normal vector $N(t)$ to the curve C and the corresponding unit normal vector $\boldsymbol{n}(u(t), v(t))$ to the surface X. The family of planes containing the normal $\boldsymbol{n}$ at the given point $\boldsymbol{P}$ cut the surface in a family of *normal section curves* for that point.

DEFINITION 3. The curve of intersection of the surface X and a plane passing through both the tangent to C at P and the normal to the X at P is called the *normal section curve* of X at P.

The curvature of all curves C with a common tangent direction at P, depends only on the angle γ between their unit principal normal vector N at P and the corresponding unit normal vector n to X. The length of the projection of the vector κN over n is called the *normal curvature* of C (Figure (4.1)).

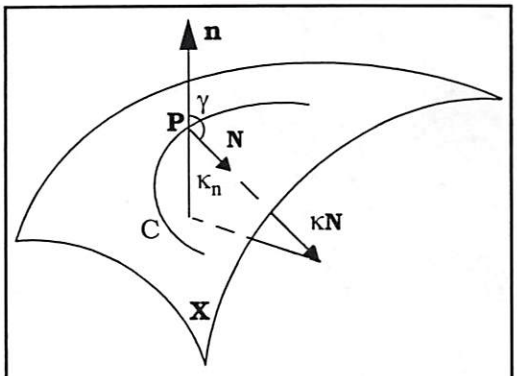


Figure 4.1. Normal Curvature at a point.

DEFINITION 4. Let C be a regular curve in X passing through P. The number $\kappa_n = \kappa\cos\gamma$ is called the *normal curvature* of C at P where κ is the curvature of C at P and γ is the angle between the unit principal normal vector N of the curve C at P and the corresponding unit normal vector n to the surface X at P.

DEFINITION 5. The *maximum normal curvature* k_1 and the *minimum normal curvature* k_2 are called the *principal curvatures* at P.

DEFINITION 6. The product of the principal curvatures is called *Gaussian curvature* and their average is called *mean curvature*.

4.3 Quantitative Analysis of Surface Curvature

At a given point for a specified direction $(\dot{u}, \dot{v})$ the normal curvature can be expressed in terms of the first and second fundamental coefficients of the surface as

$$\kappa = -\frac{L\dot{u}^2 + 2M\dot{u}\dot{v} + N\dot{v}^2}{E\dot{u}^2 + 2F\dot{u}\dot{v} + G\dot{v}^2} \qquad \text{(Eq. 4.8)}$$

The radius of curvature (the radius of the circle that best approximates the normal section curve) is $R = |\kappa|^{-1}$. The sign convention in the above equation gives positive curvature if the center of the curvature lies on the opposite side of the surface with respect to the surface normal (convex) and negative curvature when both lie on the same side (concave).

At each point Equation (4.8) associates a curvature κ with each direction $(\dot{u}, \dot{v})$ on the surface. The direction in which κ attains extremum values occur when $\frac{\partial \kappa}{\partial \dot{u}} = 0$ and $\frac{\partial \kappa}{\partial \dot{v}} = 0$. These conditions can be written as

$$(L + \kappa E)\dot{u} + (M + \kappa F)\dot{v} = 0 \qquad \text{(Eq. 4.9)}$$

$$(L + \kappa E)\dot{u} + (M + \kappa F)\dot{v} = 0 \qquad \text{(Eq. 4.10)}$$

For these equations to possess a consistent solution κ must satisfy

$$\kappa^2 - 2H\kappa + K = 0 \qquad \text{(Eq. 4.11)}$$

where the Gaussian curvature K and mean curvature H are defined by

$$K = \frac{LN - M^2}{EG - F^2} \qquad \text{(Eq. 4.12)}$$

$$H = \frac{2FM - (EN + GL)}{2(EG - F^2)} \qquad \text{(Eq. 4.13)}$$

The two solutions to the quadratic Equation (4.11) are the extremum values of the curvature as a function of direction which are called the principal curvatures. They represent the upper and lower bounds on the curvature at a given point:

$$\kappa_{max} = H + \sqrt{H^2 - K} \qquad \text{(Eq. 4.14)}$$

$$\kappa_{min} = H - \sqrt{H^2 - K} \qquad \text{(Eq. 4.15)}$$

The Gaussian and mean curvatures introduced above are readily seen to be the product and average of the principal curvatures, respectively:

$$K = \kappa_{max}\kappa_{min} \qquad \text{(Eq. 4.16)}$$

$$H = 1/2(\kappa_{max} + \kappa_{min}) \qquad \text{(Eq. 4.17)}$$

When $H^2 = K$, $\kappa_{max} = \kappa_{min}$ and κ is independent of direction. Such a point is called an *umbilical* or *spherical* point since the surface locally approximates a sphere at that point. When the principal curvatures have the same sign (as in a hill or pit) the Gaussian curvature is positive and the point is called *elliptic*. *Hyperbolic* points are those points that have negative Gaussian curvature or principal curvatures with opposite signs (as in saddle points). When either one of the principal curvatures is zero the Gaussian curvature is zero (as in ruled or cylindrical surfaces). Surfaces that have zero Gaussian curvature everywhere are *developable*, i.e. they can be laid flat on a plane without stretching or tearing. If both principal curvatures are zero the point is called *planar* but if only one of them is zero it is called *parabolic*.

4.4 Curvature Maps

A convenient method for displaying the variation of a scalar quantity over a surface is by means of a color-coded map, an even gradation of color corresponding to a range of values for that quantity.

The three primary color components of each pixel are determined by applying some linear mapping to the value of the scalar quantity associated to that pixel. The linearity of the mapping is desired because it makes computations such as color interpolation easier. Suppose we wish to interpolate from color A to color B, we might write $C = (1-\alpha)A + \alpha B$ and sweep α from 0 to 1. This is the typical way that Gouraud and Phong shading are implemented for rendering of triangulated surfaces. We would like equal increments of α to result in steps of C that were of perceptually equal size. This simple interpolation scheme works only if the mappings between the value of the scalar quantity and the red, green and blue color components of each pixel are linear.

By using the same mapping function for the three color components the resulting map will be a grey scale image with different shades of grey corresponding to different values of the scalar quantity.

We have generated different curvature maps for several objects including synthetic phantoms and real human brain. The surface that we start with is reconstructed from the slice images of the objects using the tessellation algorithm introduced in Chapter 2. To calculate the curvatures we fit a local quadratic model to the surface patch in the neighborhood of each node of the tessellated surface as was described in Chapter 3.

Following the notation used in Chapter 3, let $\boldsymbol{X}(u, v)$ be a parametrization at a point $\boldsymbol{X}_i \in S$ of a surface S given by

$$\boldsymbol{X}(u, v) = \boldsymbol{X}_i + u\boldsymbol{P}_i + v\boldsymbol{Q}_i + ([u, v]C_i[u, v]^t)\boldsymbol{n}_i \qquad \textbf{(Eq. 4.18)}$$

The first and second fundamental coefficients of the surface can be calculated from this representation using the Equations (4.2)-(4.7). According to the *equations of Weingarten* [22] the principal curvatures κ_1 and κ_2 of the surface are defined as the eigen values of the matrix

$$-A = \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \qquad \textbf{(Eq. 4.19)}$$

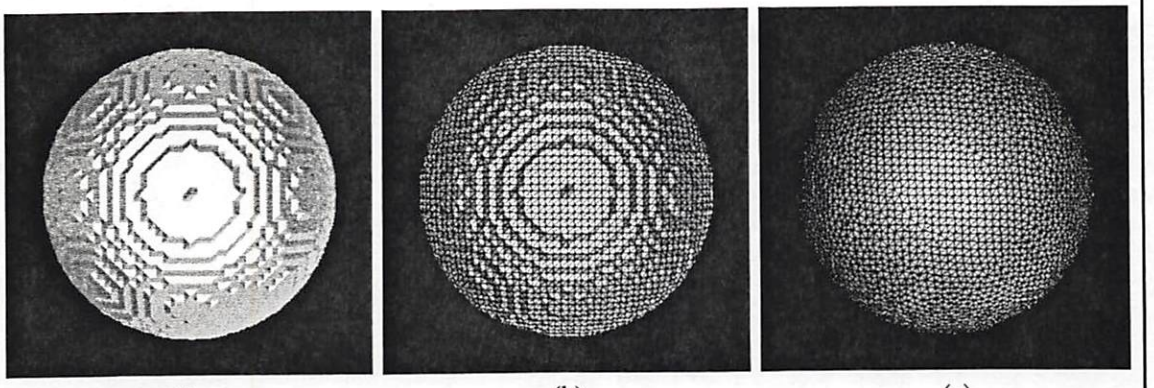
and it can be shown [33] that this matrix is equal to the coefficient matrix of the second order terms in Equation (4.18), i.e.

$$-A = 2C_i \qquad \textbf{(Eq. 4.20)}$$

So, in order to calculate the minimum and maximum curvatures at each point we find the eigenvalues of the corresponding matrix C_i at that point, and then we use Equations (4.16) and (4.17) to calculate the mean and Gaussian curvatures.

4.5 Experimental Results

To examine the accuracy of this method in calculating different curvatures of a surface we have tested the algorithm on several phantoms. All phantoms were designed to provide the same type of data set that we expected from a real cortical surface. To generate a phantom first, a binary volume composed of a stack of cross-sectional images is created. This volume represents a discretized approximation of an object with a well defined analytical description that can be used to verify the results of our calculations. The phantoms are then tessellated using the algorithm described in Chapter 2. The tessellated object is then slightly deformed by adjusting the location of its vertices to reduce the difference between



(a) (b) (c)

Figure 4.2. Sphere phantom. (a) Binary volume. (b) Tessellated Surface. (3) The nodes in the triangulated sphere are repositioned to new locations that conform with the analytical description of an exact sphere.

the discretized shape and the continuous form of the object. In contrast to a discretized volume representation of objects in a rectangular grid, in a mesh representation the location of nodes does not necessarily have to be on the grid points. Figure (4.2.a) shows a tessellated discretized spherical phantom in rectangular grid. Figure (4.2.b) shows the same phantom, and the modified tessellated phantom that conforms with an analytically modeled sphere is displayed in Figure (4.2.c).

Figure (4.3) shows minimum, maximum, mean and Gaussian curvature maps of a sphere. A grey scale colormap is used to display the variation of the curvatures. As can be easily

verified, the difference between calculated and mathematically derived values of the curvatures of the sphere are small. Given that the values of all curvatures should be constant (minimum curvature: $1/R$, maximum curvature: $1/R$, mean curvature: $1/R$, Gaussian curvature: $1/R^2$, where R is the radius of sphere) a close investigation of the variation pattern of these curvatures indicates that the error is primarily due to the deviation of the shape of the phantom from a real sphere.

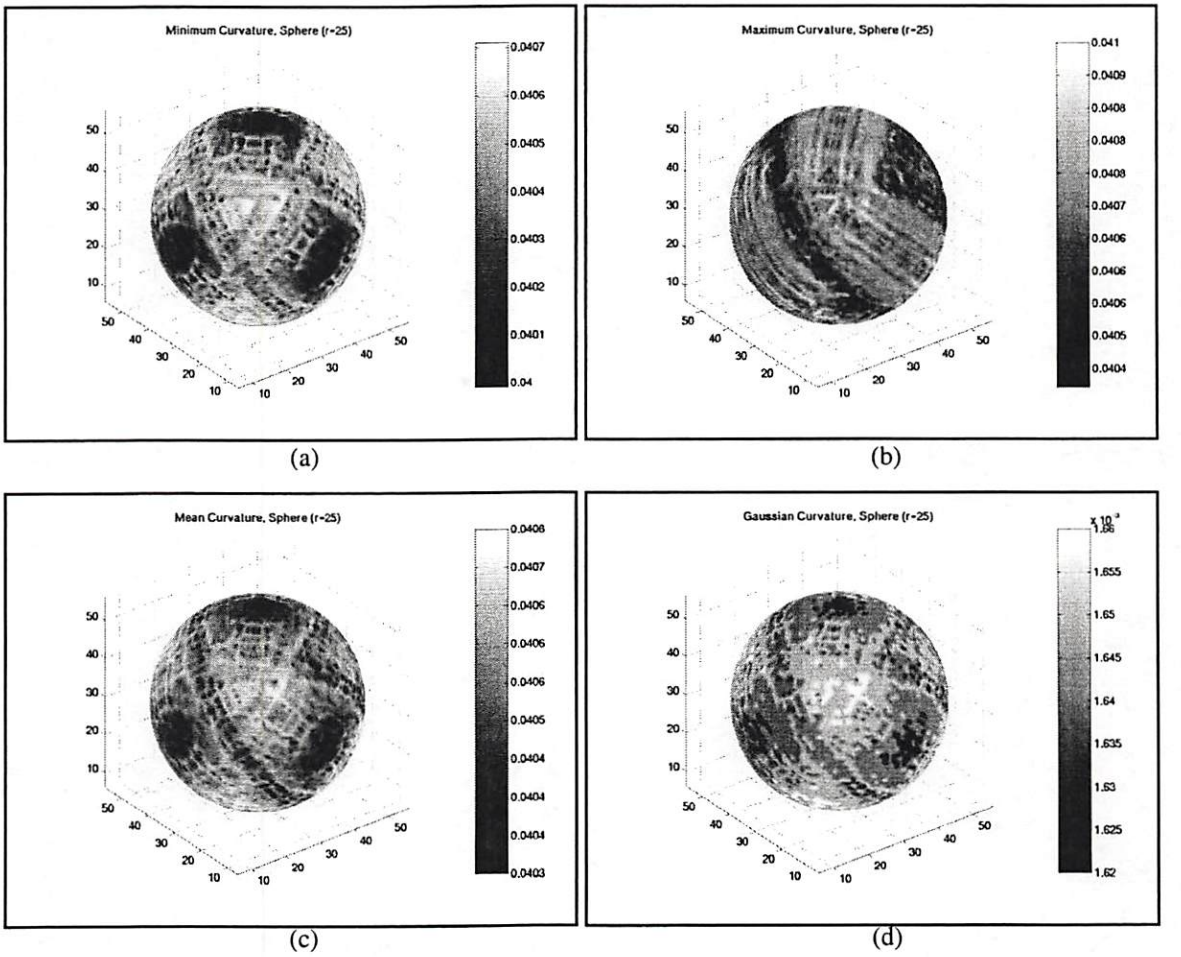
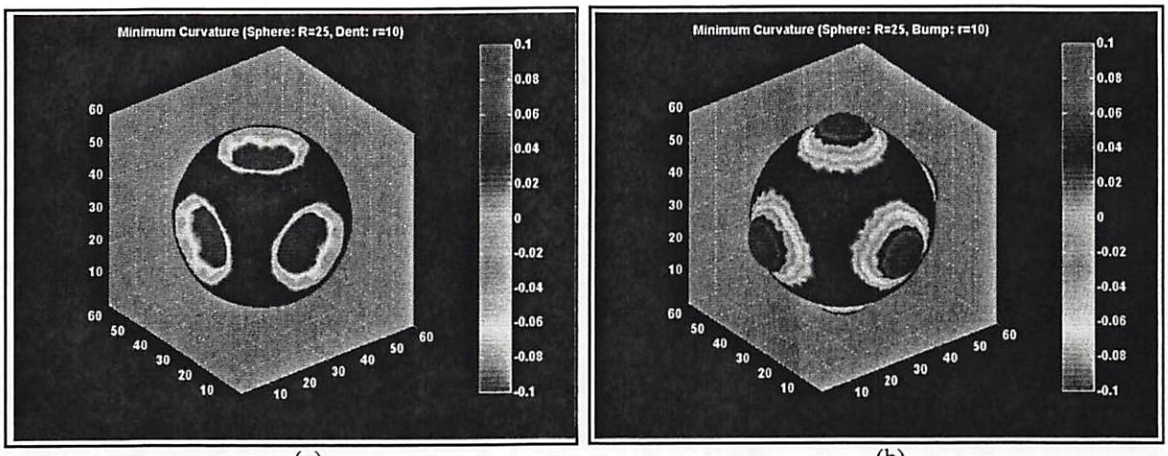


Figure 4.3. Curvature maps for a sphere with a radius of 25 units. (a) Minimum curvature map. (b) Maximum curvature map. (c) Mean curvature map. (d) Gaussian curvature map.

The second set of experiments were performed on phantoms specifically designed to simulate cortical folds. To study the variation of different curvatures in the neighborhood of a depression and a protuberance in a surface we created two new phantoms by making

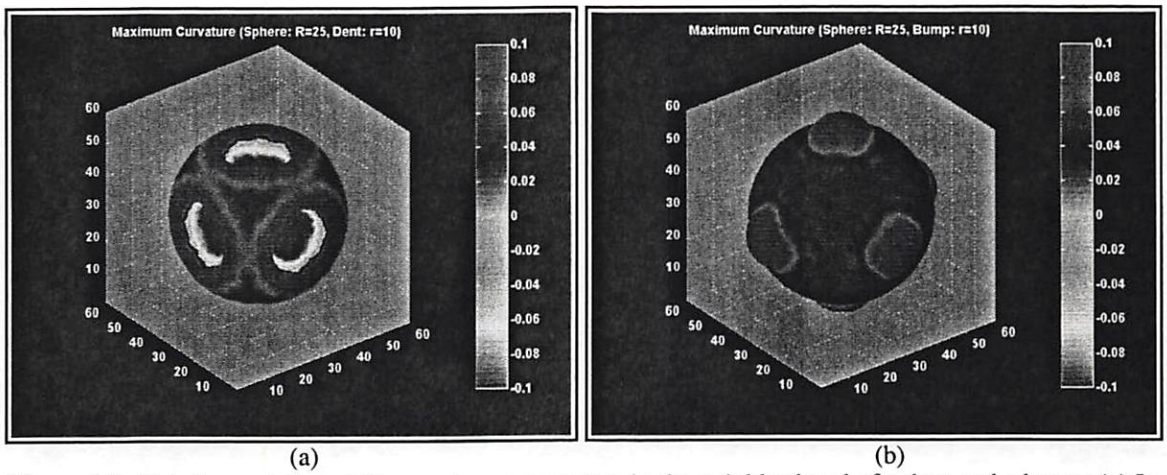
several dents and bumps in our spherical phantom. In Figures (4.4) and (4.5) we have displayed the minimum and maximum curvature maps of these two phantoms, respectively. Comparing the curvature maps of the indented sphere with those of the bumpy one, we can observe that curvature values in the neighborhood of a dent are always negative whereas those corresponding to a bump are positive. One can justify this fact by noting that the normal curvature is proportional to $Cos\gamma$ where γ is the angle between the unit principal normal vector of a curve passing through a point and the corresponding unit normal vector to the surface at that point [22]. Since a dent and a bump of equal size are mirror images of each other with respect to the surface of the sphere, the principal normal vector of a pair of corresponding curves passing through a pair of corresponding points - one on a bump the other on a dent - will point in opposite directions. However, since the normal vector to the surface is always pointing outward there will be a 180 degree difference between the values of the angle γ for a dent and a bump which accounts for the negative sign.

We can also verify that the absolute value of the curvature depends on the size of the bump or dent. The mathematical explanation of this fact is based on the notion of radius of curvature. At each point of the surface one may fit circles to the point and two of its neighbors such that these circles lie in the normal sections of the surface. The maximum and minimum curvature of the surface at a given point is equal to the inverse of the maximum and minimum radii of such circles passing through that point. As the size of the bump or dent becomes smaller the radius of the circles that can be fit to it decreases, causing an increase in the maximum and minimum curvature and consequently the mean and Gaussian curvatures.



(a) (b)

Figure 4.4. Relative variation of the minimum curvature in the neighborhood of a dent and a bump. (a) In the neighborhood of a dent, the minimum curvature takes negative values. (b) In the neighborhood of a bump, the minimum curvature takes positive values. Minimum curvature is a good feature for locating dents.



(a) (b)

Figure 4.5. Relative variation of the maximum curvature in the neighborhood of a dent and a bump. (a) In the neighborhood of a dent, the maximum curvature takes negative values. (b) In the neighborhood of a bump, the minimum curvature takes positive values. Maximum curvature is a good feature for locating bumps.

Comparing the minimum and maximum curvatures in the neighborhood of a bump and a dent as demonstrated in Figures (4.4) and (4.5), one can observe that maximum curvature is a perfect feature for locating "hills" in a surface, and similarly minimum curvature for identifying "valleys". In the neighborhood of a hill, all the points have positive normal curvatures. Using the fact that maximum curvature is the maximum normal curvature of all curves passing through a point, we can conclude that maximum curvature takes the largest positive value on top of a hill. By similar reasoning, the minimum curvature takes the most negative value at the bottom of a valley. We can use this property of maximum and minimum curvature to locate sulci and gyri on the cortical surface.

We have calculated the minimum, maximum, mean and Gaussian curvature of a typical human cortical surface. The curvatures have been plotted both on the original extracted cortical surface of our sample data and its inflated version. In order to generate a reasonable curvature map for the cortex we have to ignore a few out of range values resulting from computational errors. The curvature of these points are replaced by the average value of the curvature of the points in their neighborhood. To determine the limit of acceptable values, we use the sorted histogram of the quantity and pick a threshold value such that more than 99% of points have values less than the threshold.

From the Figures (4.6) and (4.7), we can see that minimum and maximum curvatures can distinctly identify the location of sulci and gyri on the cortex. Large negative values of the minimum curvature (dark grey areas) correspond to sulci and large positive values of the maximum curvature (light grey areas) match with the gyri.

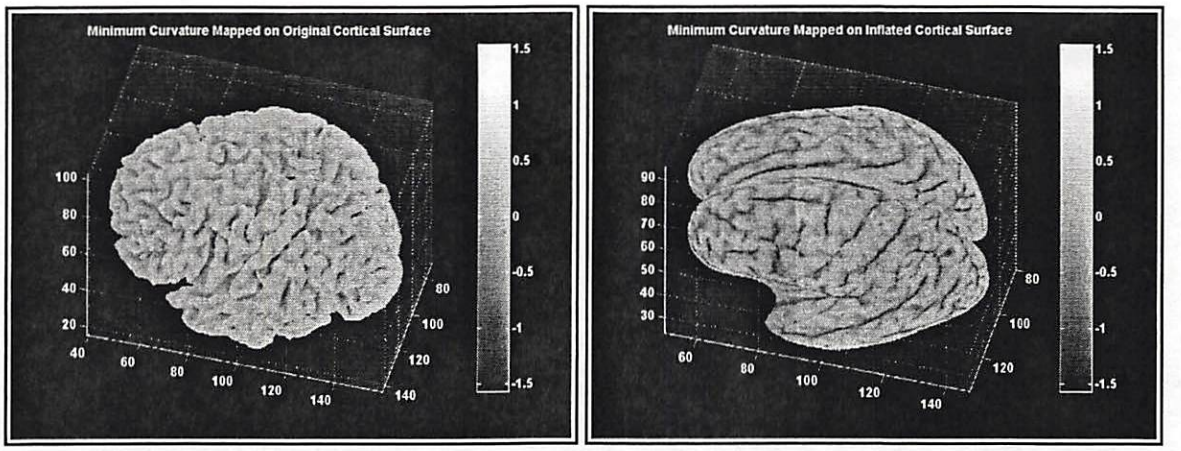


Figure 4.6. Minimum curvature of human cortical surface mapped onto: (a) Original brain. (b) Inflated brain.

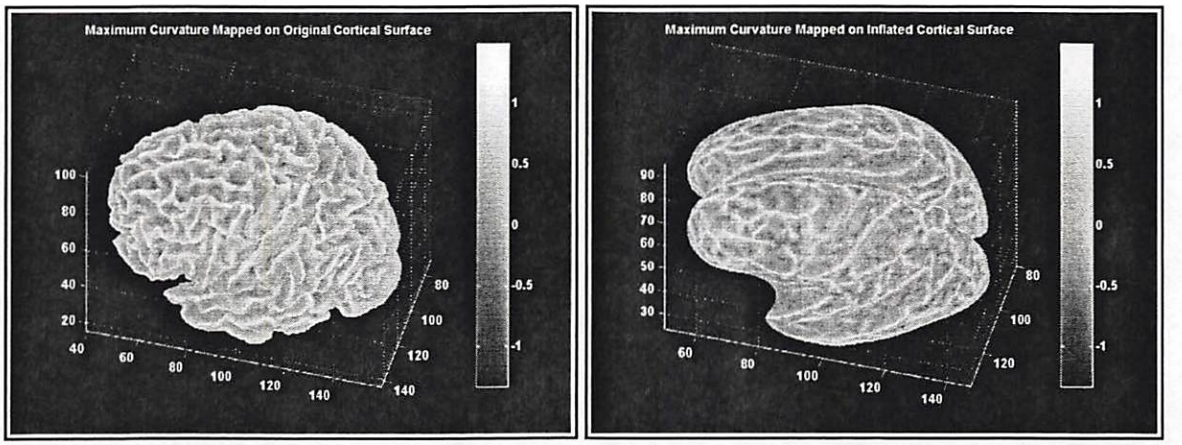


Figure 4.7. Maximum curvature of human cortical surface mapped onto: (a) Original brain. (b) Inflated brain.

The mean curvature (Figure (4.8)) by itself is particularly informative for ascertaining both types of major cortical folds. It takes positive values where the cortex is folded inward along the crown of a gyrus, and negative values where the cortex is folded outward along the fundus of a sulcus.

Finally, Figure (4.9) displays the Gaussian curvature of the human cortical surface. Clearly, the Gaussian curvature of brain cortex is not a constant value equal to zero; hence it is not a developable surface. However, its mean is close to zero and its variance is very small.

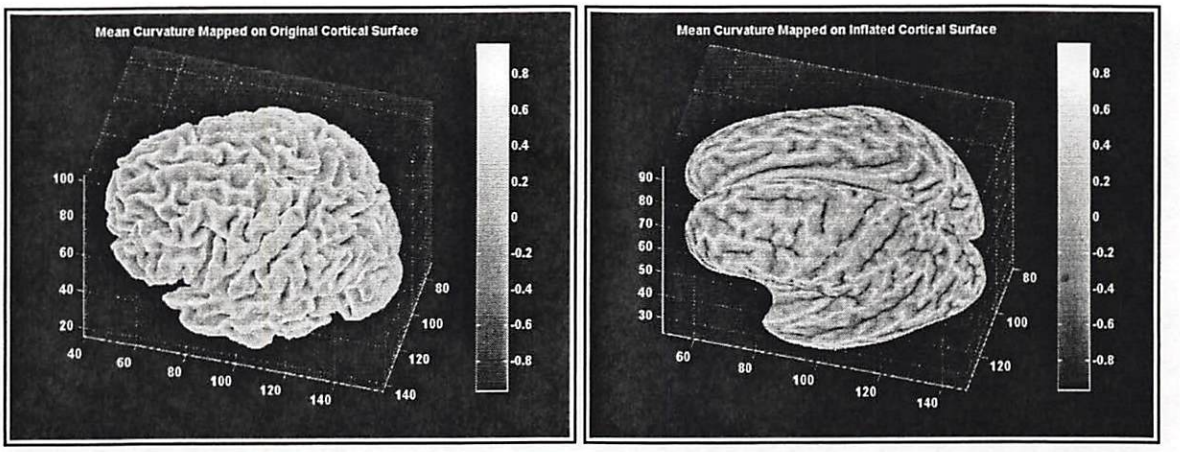


Figure 4.8. Mean curvature of human cortical surface mapped onto: (a) Original brain. (b) Inflated brain.

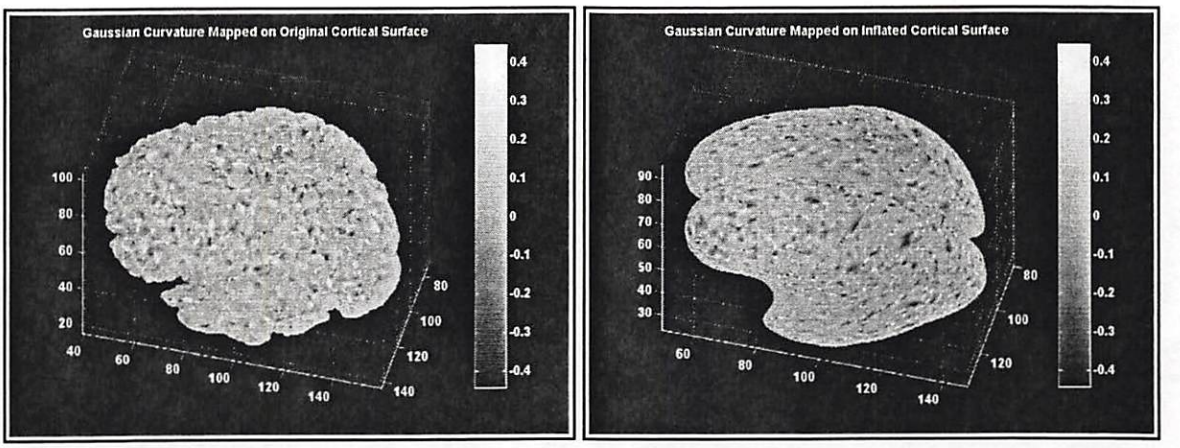


Figure 4.9. Gaussian curvature of human cortical surface mapped onto: (a) Original brain. (b) Inflated brain.

CHAPTER 5 *Finding the Flat Map of Human Cortical Surface*

In the last two decades, a growing interest has developed among neurologists and engineers in studying the problem of unfolding the cortical surface and generating flat maps of the human brain. A central problem in the study of cortical organization and function is the difficulty in analyzing regions buried within the deep and irregularly convoluted sulci of the cortex. The difficulty is not simply in visualization, because buried regions can be exposed by taking two dimensional slices through the cortex using various histological or noninvasive imaging techniques. Rather, the main challenges are studying the functional architecture of cortex and measuring detailed maps of neural activity on the complex curved surface of the brain.

This chapter is intended to provide a review of existing cortex unfolding and flattening methods and to establish the motivation for the design of our geometric distortion minimization techniques for generating flat maps of the human brain. We start with a brief overview of the mathematical definitions of three important mappings.

5.1 Mappings

The first systematic investigation of the problem of surface mapping is attributed to the construction of the globe map for nautical or other purposes. Of particular theoretical and practical interest are mappings that preserve certain geometric properties. The most important types are isometric or length-preserving mappings, conformal or angle-preserving mappings and equiareal or area-preserving mappings. All these mappings

belong to a general class of mappings called *diffeomorphisms*. A differentiable map φ from a portion X of a regular surface onto a portion Y of a regular surface is called *diffeomorphism* if its inverse φ^{-1} exists and is differentiable.

***Isometric mapping*.** An *isometric mapping* or *isometry* is a diffeomorphism $\varphi : X \to Y$, such that the length of any arc on Y is the same as that of its inverse image on X.
A diffeomorphism $\varphi : X \to Y$ is isometric if and only if at corresponding points of X and Y, when referred to the same coordinate systems, the coefficients of the first fundamental forms on X and Y are equal [22][36].

***Conformal mapping*.** A *conformal* or *angle-preserving* mapping is a diffeomorphism $\varphi : X \to Y$, such that the angle of intersection of every arbitrary pair of intersecting arcs on Y is the same as that of the inverse images on X at the corresponding point. The angle between two intersecting curves is, by definition, the angle between the tangents to these curves at the point of intersection.
A diffeomorphism $\varphi : X \to Y$ is conformal if and only if, when on X and Y the same coordinate systems have been introduced, the coefficients of the first fundamental forms are proportional [22][36].

***Equiareal mapping*.** An *equiareal* or *area-preserving* mapping is a diffeomorphism $\varphi : X \to Y$, such that every part of X is mapped onto a part of Y which has the same area.
A diffeomorphism $\varphi : X \to Y$ is equiareal if and only if at corresponding points of X and Y, when referred to the same coordinate systems, the discriminants of the first fundamental forms on X and Y are equal. The discriminant of the first fundamental form is defined as $EG - F^2$, where E, F, and G are the first fundamental coefficients [22][36].
It can be proved that every isometric mapping is also conformal and equiareal, and every equiareal and conformal mapping is isometric. It is of basic importance that there is no

difference in the measurement of lengths, angles, and areas in isometric surfaces, although the surfaces, when considered from the embedding space may have an entirely different geometric shape.

The concept of isometric mapping is usually used in the same sense as *bending deformation*, a surface deformation that preserves the length of every arc on the surface. Unfolding is considered as a special type of *bending deformation*. Surfaces which can be transformed into each other by bending are called *applicable*.

Two distinct points on a curved surface can be connected by many different paths on the surface, and each path has in general a different length. Those paths corresponding to minima of the path length function are analogous to straight lines in Euclidean space and are known as *geodesics*. A bending maps geodesics to straight lines.

The *geodesic distance* between two points is defined to be equal to the length of the geodesic connecting them. Geodesic distance depends only on the first fundamental form of the surface and it remains unchanged under bending. Properties that remain unchanged under bending are called *bending invariants*. Other examples of bending invariants are surface area and Gaussian curvature. Counter examples are Euclidean distance, enclosed volume and mean curvature.

The properties of a surface which depend on the first fundamental form only and are bending invariant are called *intrinsic properties*. Intrinsic properties are invariant under transformations which do not stretch or tear the surface. Correspondingly, there are properties of the surface that depend on the particular configuration of the surface in the space. These properties are called *extrinsic properties.* Thus a sheet of paper has the same intrinsic properties whether it is flat or crumpled but not the same extrinsic properties.

5.2 Survey of Current Techniques

We have categorized published works on unfolding and flattening of the cortical surface into two general categories of mechanical and geometrical approaches, which are based on the underlying model they use for representing the brain surface.

5.2.1 Mechanical Model Based Flattening Algorithms

In this class of algorithms the material of the brain is modeled as an isotropic elastic substance that is moving under a system of forces. Depending on the definition of the system of forces and the final equilibrium state of the flattened brain, different physical interpretations can be made for the behavior of algorithms in this class.

The simplest solution in this category, proposed by Dale and Sereno, is a shrink-wrap method for blowing up the elastic surface of cortex by simulating the way a folded balloon is blown up [18]. The basic idea behind this methods is to start with a three-dimensional wire frame reconstruction with the correct topology and then gradually deform the shape of the surface by rubber sheet transformations to conform to the cortical sheet. The location of each vertex on the surface is updated iteratively according to elastic forces between neighboring vertices and repulsive and attractive forces along local surface normals which depend on the MRI data at the vertex. The criteria used in this unfolding scheme are not optimal in terms of preserving any metric property of the resulting map. They preserve only local properties of the surface and therefore do not rule out large-scale distortions caused by locally correlated errors.

An improved version of this method has been recently proposed [27], in which the resulting map is forced to have a minimum distortion in some geometric measures of the surface. In this method the external force is an unfolding force that acts to flatten the surface by making each node closer to being coplanar with its neighbors, and at the same time, to resist internal forces that are attempting to maintain the intrinsic curvature of the

original surface. The internal forces model the elastic properties of the surface. They act to restore the length between adjacent nodes to their reference values in order to preserve surface area as well as distances within the cortex.

With a similar approach, Van Essen and his team have developed algorithms for creating two-dimensional maps of the cortex by modeling it as an elastic sheet which is continuously deformed to a planar configuration with minimal energy. The energy is defined as a measure of the difference between local lengths and angles on the two dimensional map and the corresponding lengths and angles in the three-dimensional surface. The first version of this algorithm started from an arbitrary planar configuration with the same topology as cortex, possibly with different geometry. The map was then subjected to random perturbation and configurations were selected that had progressively lower energy, until a mapping of minimum overall distortion was asymptotically approached [9].

Later, in 1995, Carman *et al.* [11] enhanced the cortex model to include deformations generated by a system of forces. In the new model, internal forces acted to preserve the topology and geometry of the sheet while external forces acted to unfold and flatten it. The unfolding force is designed to drive the overall surface into a planar configuration and to act against internal forces attempting to maintain intrinsic curvature or local folding. The internal forces are the longitudinal and torsional forces, which together model the elastic properties of the surface. The starting configuration for the dynamic unfolding process, which also provided the reference geometry, was a wire-frame representation of the reconstructed brain surface.

Drury, *et al.* [23] have recently extended this method by incorporating the concept of multiresolution processing. By making a global shape change on a coarse-resolution representation of the surface and then transferring it to finer-resolution representations,

which are later subjected to more localized shape refinements, they reduce the computational demands of the unfolding process.

This enhanced method represents some improvements over the previous ones developed by this group in terms of accuracy of results and speed of processing. The criteria used for creation of the flat map is to minimize the distortion in local lengths and angles with reference to those in some wire-frame representation of reconstructed brain surface. Because of this, although this method is not explicitly formulated to find an isometric mapping between the brain surface and plane, the outcome is equivalent.

5.2.2 Geometrical Model Based Flattening Algorithms

In the second class of algorithms the cortical surface is considered to be a geometric entity not a mechanical object. In this geometrical model, minimizing the distortion in some geometric measure of interest replaces the objective of finding the equilibrium state of an elastic sheet subject to a system of forces. There are four measures one may want to retain in the flat map of the cortex: the neighborhood relationship between different regions of brain, the area of these regions, the length of curves connecting a single point to the other points on the surface, and the angles between these curves. The transformations which preserve these measures are called topological, equiareal, isometric and conformal, respectively. Although there have been some reports on equiareal mappings of rat brain surface to the plane [1], isometric mappings appear to be the preferred subject of study for many researchers.

Seeking the optimal preservation of the metric structure of cortex, Schwartz *et al.*, [48] formulated a variational problem to maximize the goodness of fit of the "distance matrix" of the original 3-D model and the flattened model of the brain. The distance matrix is defined as the matrix of all possible inter-point distances. In a polyhedral model of the surface, distances must be defined as minimal geodesic distances along paths embedded in the surface so that there is a match between straight lines in the plane and distances

defined on the polyhedron. The goal is to determine a set of points in the plane such that the corresponding distance matrix in the plane provides a best fit, in the least squares sense, to the distance matrix of the original surface. This is done by applying a Newton-Raphson gradient descent algorithm on the difference between the three-dimensional (geodesic) distance matrix and that of a corresponding set of initially random two-dimensional points.

The resulting flat map is optimal in the sense that it is derived from a variational principle that optimizes the overall fit between the curved and planar surfaces. However the major drawback of this algorithm is the difficulty of finding the minimal distances. Their suggested solution to the problem of finding geodesic distances on a surface has an exponential complexity and the requirements of performing this procedure in terms of both memory and time becomes prohibitive as the level of complexity of the surface increases.

So far, all the methods that we have referred to, in spite of their differences in approach are similar in the sense that they start from the polyhedral model of the cortex and base all the calculations on that model. Alternatively, one can start from a parametric model of the cortex as suggested by Davitzikos. In [19][20], he models the outer cortex as a thick sheet of grey matter which is deformed toward its central layer as an equilibrium is reached between two sets of opposing forces: external forces trying to unfold the surface and internal forces acting as elastic resistance which preserves the connectivity of the surface. The equilibrium configuration of the surface, which is the solution to the associated elasticity differential equation, provides a numerical description of cortex as a function of two parameters. Davitzikos then uses this parametric representation as the initial value for a fixed point algorithm whose final answer will be a parameterization isometric to the plane. This algorithm uses the fact that first fundamental coefficients for the plane are $E = G = 1$ and $F = 0$. Instead of trying to reparameterize the entire surface $X(u, v)$ to

globally change these values, only the isoparametric curves (*Constant-u* curves, and *Constant-v* curves) are reparameterized to become unit speed curves (satisfying $E = G = 1$) and also intersect each other at right angles (satisfying $F = 0$).

5.3 Flattening As An Optimization Problem

In the process of developing the flat map of a particular surface the first question to be answered is whether the surface is isometric to the plane. A curved surface that can be brought into isometric correspondence with a plane is said to be *developable*. Since an isometric mapping does not change the intrinsic geometry of a surface, the fact that the Gaussian curvature of plane is zero everywhere implies that the Gaussian curvature of a developable surface must also be identically zero.

Most surfaces are not developable, for example a sphere has a constant Gaussian curvature and can not be mapped isometrically into the plane with zero error. This is the so-called map-maker's problem, the problem of portraying the surface of the earth on a flat plane. Given that the earth is approximately spherical, some distortion of conformality, distance, direction, scale, and area always results from projecting the earth onto a plane. Some projections minimize distortions in some of these properties at the expense of maximizing errors in others. Some projection are attempts to moderate distortion in all of these properties.

The results of our experiments with human brain (Chapter 4), as well as the reports of other researchers conducting experiments on macaque monkey [47][33], indicate that the Gaussian curvature of cortical surface is not zero; however, it is relatively constant over most of the surface. In contrast to the sphere, this result suggests that flattening of cortex with some small distortion is possible, with larger than average errors occurring in the neighborhood of sulci and gyri where the Gaussian radius of curvature is larger than average [47].

The work we will describe in this section is an effort to find a practical approximate solution to a problem for which no exact solution exists. Formally, our approach entails the specification of the flat map of cortex as a solution to a constrained optimization problem. We formulate the constraints as non-negative energy functions in three dimensional space with zeros at point-configurations satisfying the constraints. We refer to the constraint functions as energy functions not because they model the energy of actual physical systems, but because they play a role similar to that of physical energy functions. Intuitively, energy constraints may be viewed as forces that pull the surface into the desired configuration and hold it there, although they are not necessarily intended to be physically realistic forces.

We are seeking an optimal configuration of points which minimizes the integral of a surface metric subject to some geometric constraints. An ideal surface metric for our purpose is one that reaches its extremum value when the cortical surface deforms to an unfolded configuration. The role of geometric constraints is to impose the requirements for an isometric mapping. One can choose the preservation of geodesic distances as a constraint directly characterizing isometry. However, computing geodesic distances on an arbitrary manifold is a difficult problem and so far none of the proposed solutions are computationally tractable for huge data sets such as the human brain [63][39][21]. Alternatively, we have chosen to use constraint functions that preserve areas and angles, thus imposing equiareal and conformal mapping requirements. This will serve our purpose because, as we mentioned earlier in this chapter, if we find a transformation that is both equiareal and conformal, it can be shown to be isometric.

5.3.1 Objective Function: Minimizing "Convexity"

By defining a measure for quantifying the folding pattern of a surface one can numerically evaluate the deviation of a surface from a flat configuration. Ideally, such a measure should be able to discriminate between surface folds based on their size and their folding

direction. Mean curvature captures the folding pattern of a surface but it is equally sensitive to large and small folds, yet it distinguishes between fold-in and fold-out and presents the difference with a sign change (Figure (5.1.a)).

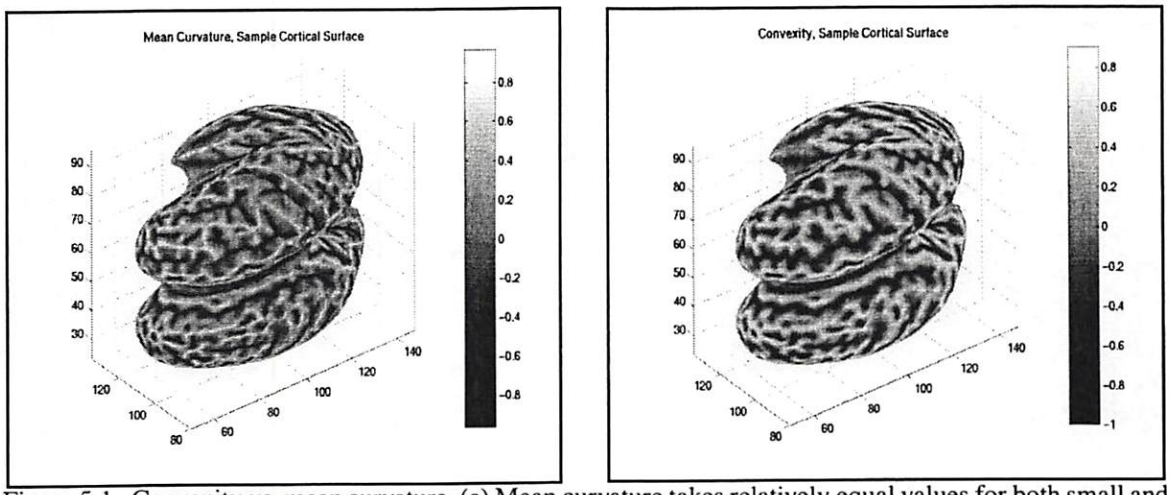


Figure 5.1. Convexity vs. mean curvature. (a) Mean curvature takes relatively equal values for both small and large folds. (b) Convexity takes high values for large folds and is almost insensitive to small folds.

Computational cost can be considered a drawback for mean curvature if it is used in a cost function of an optimization problem dealing with large data sets. As explained in Chapter 4, the calculation of mean curvature at each point of the surface involves finding the coefficients of a local polynomial patch through solving an overdetermined system of linear equations, a process requiring the computationally demanding task of calculating the pseudo-inverse of a matrix.

Motivated by the definition of locally convex surfaces [22] we have come up with an alternative measure for determining the convexity of a surface that is fast and easy to calculate for tessellated surfaces.

DEFINITION. A surface X is *locally convex* at a point P if there exists a neighborhood N of P (subset of X) such that N is contained in one of the closed half-spaces determined by the plane tangent to X at P. If in addition, N has only one common point with the tangent plane, then X is called *strictly locally convex* at P.

We define the *convexity* of a surface S at a point X_i to be equal to the average of the inner products of the unit surface normal vector n_i at point X_i and the unit edge vectors X_{ij} linking the point X_i to its neighbors X_j, or equally the average of the cosines of the angles between the normal and the edges (Figure (5.2)).

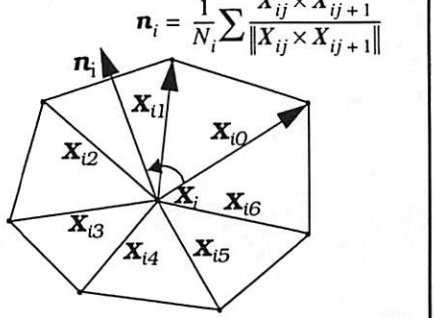


Figure 5.2. X_i, a typical node of a tessellated surface. n_i represents surface normal and X_{ij} represent the edges.

$$Conv_i = \frac{1}{N_i} \sum_{j=0}^{N_i - 1} n_i \cdot \frac{(X_j - X_i)}{\|X_j - X_i\|} \qquad \text{(Eq. 5.1)}$$

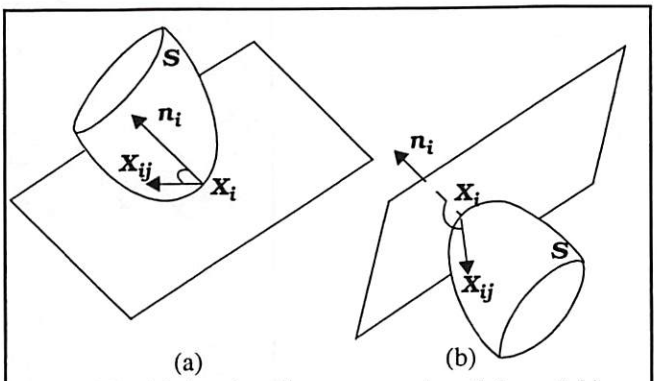


Figure 5.3. (a) At a locally convex point all the neighbors are above the tangent plane. (b) At a locally concave point all the neighbors are below the tangent plane.

Assuming that the surface normal is always pointing outward, we can verify that if the surface is locally convex at a point, all the neighbors of that point are above the tangent plane; therefore, the angles between the normal and edges are all acute and convexity is positive (Figure (5.3.a)). On the other hand, if the surface is locally concave at a point, all the neighbors of that point are below the tangent plane, hence all the angles are obtuse and convexity is negative (Figure (5.3.b)). For a locally planar portion of a surface all the angles are 90° and the convexity is zero. With this definition the convexity for a saddle point may be zero; however, the convexity must be nonzero in a local neighborhood so that a continuous surface containing saddle points is not in a minimum convexity configuration.

Based on this convexity measure we have defined an energy functional I_c that quantifies the deviation of a surface from a planar configuration and is zero only when the surface is a flat plane.

$$I_c = \frac{1}{N}\sum_{i=0}^{N-1} Conv_i^{\,2} = \frac{1}{N}\sum_{i=0}^{N-1}\left[\frac{1}{N_i}\sum_{j=0}^{N_i-1} \boldsymbol{n}_i \cdot \frac{(X_j - X_i)}{\|X_j - X_i\|}\right]^2 \qquad \text{(Eq. 5.2)}$$

In Equation (5.2) we have used the symbols $X_i : i = 0, 1, \ldots, N-1$ (N is the total number of vertices of the mesh) to refer to the vertices of the mesh representation of the surface, and $\boldsymbol{n}_i$ to refer to the surface normal at point X_i. We assume that each point X_i has a total of N_i neighbors that we represent by $X_j : j = 0, 1, \ldots, N_i - 1$. I_c is proportional to the average squared convexity over all points on the surface. The reason for using the squared value is to treat convex and concave points equally. The global minimum of this function is where the surface reaches a planar configuration.

In order to minimize I_c, we need to calculate the gradient of I_c with respect to the vector of vertex positions $X = (X_0, X_1, \ldots, X_{N-1})^t$ where $X_i = (X_{ix}, X_{iy}, X_{iz}) : i = 0, 1, \ldots, N-1$. For this purpose we take the derivative of I_c with respect to X_i,

$$\frac{\partial I_c}{\partial X_i} = \frac{2}{N}\sum_{j=0}^{N_i-1}\left\{\left(-\frac{1}{N_i}Conv_i\right)\left[\frac{1}{\|X_j - X_i\|}\boldsymbol{n}_i - \frac{1}{\|X_j - X_i\|^3}(\boldsymbol{n}_i \cdot (X_j - X_i))(X_j - X_i)\right]\right\} + \qquad \text{(Eq. 5.3)}$$

$$\left\{\left(\frac{1}{N_j}Conv_j\right)\left[\frac{1}{\|X_j - X_i\|}\boldsymbol{n}_j - \frac{1}{\|X_j - X_i\|^3}(\boldsymbol{n}_j \cdot (X_j - X_i))(X_j - X_i)\right]\right\}$$

In the above equation and all future relations that we will derive for the derivatives of our energy functionals, we ignore the dependency of the unit surface normal on vector of vertex positions and therefore we assume that $\frac{\partial \boldsymbol{n}_i}{\partial X_k} = \boldsymbol{0}$, for all $i = 0, 1, \ldots, N-1$ and $k = 0, 1, \ldots, N-1$.

5.3.2 Constraint Function: Preserving the Area

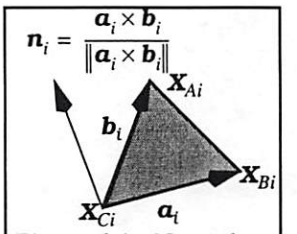


Figure 5.4. Normal to the plane of a triangle.

In order to define the constraint energy functional corresponding to the preservation of area, we consider the i^{th} triangle in the surface tessellation with unit normal vector $\boldsymbol{n}_i$, and edges $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ connecting the vertex X_i to two of its neighbors. The area of this triangle is given by half the norm of the cross product of the edges $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$.

The areal constraint function I_s is defined to penalize the difference between the current area and the original area occupied by each triangle. It is proportional to the average of the squared difference in the areas of the triangles constructing the mesh in their current forms and their original shapes.

$$I_s = \frac{1}{T}\sum_{i=0}^{T-1}[S_i^n - S_i^0]^2 = \frac{1}{T}\sum_{i=0}^{T-1}\left[\frac{1}{2}\left\|(\boldsymbol{a}_i^n \times \boldsymbol{b}_i^n)\right\| - \frac{1}{2}\left\|(\boldsymbol{a}_i^0 \times \boldsymbol{b}_i^0)\right\|\right]^2 \qquad \text{(Eq. 5.4)}$$

In this relation T refers to the total number of triangles in the mesh and n indicates the iteration number, with $n = 0$ being the starting value corresponding to the original cortical surface. In order to get a closed form expression when calculating the derivative of the area of each triangle, we use the following relation:

$$S_i = \frac{1}{2}\left\|\boldsymbol{a}_i \times \boldsymbol{b}_i\right\| = \frac{1}{2}(\boldsymbol{a}_i \times \boldsymbol{b}_i)\cdot \boldsymbol{n}_i \qquad \text{(Eq. 5.5)}$$

where $\boldsymbol{n}_i$ is the unit vector normal to the plane of the triangle. To demonstrate the functional dependence of I_s on the position of the vertices of the surface one can express $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ as functions of the vertex X_i and its neighbors as

$$I_s = \frac{1}{T}\sum_{i=0}^{T-1}\left(\frac{1}{2}(((X_{Bi}^n - X_{Ci}^n)\times(X_{Ai}^n - X_{Ci}^n))\cdot \boldsymbol{n}_i^n) - S_i^0\right)^2 \qquad \text{(Eq. 5.6)}$$

where we have renamed X_i to X_{C_i} and used Figure (5.4) as reference to describe the sides of the triangle in terms of its vertices.

Taking the partial derivative of I_s with respect to k^{th} vertex results in

$$\frac{\partial I_s}{\partial X_k^n} = \frac{2}{T}\sum_{i=0}^{T}(S_i^n - S_i^0)\frac{\partial S_i^n}{\partial X_k^n} \qquad \text{(Eq. 5.7)}$$

In order to calculate $\frac{\partial S_i^n}{\partial X_k^n}$ we use the following relations from vector analysis [56]

$$\frac{\partial}{\partial u}(\boldsymbol{A}\cdot\boldsymbol{B}\times\boldsymbol{C}) = \boldsymbol{A}\cdot\boldsymbol{B}\times\frac{\partial \boldsymbol{C}}{\partial u} + \boldsymbol{A}\cdot\frac{\partial \boldsymbol{B}}{\partial u}\times\boldsymbol{C} + \frac{\partial \boldsymbol{A}}{\partial u}\cdot\boldsymbol{B}\times\boldsymbol{C} \qquad \text{(Eq. 5.8)}$$

$$\boldsymbol{A}\cdot(\boldsymbol{B}\times\boldsymbol{C}) = \boldsymbol{B}\cdot(\boldsymbol{C}\times\boldsymbol{A}) = \boldsymbol{C}\cdot(\boldsymbol{A}\times\boldsymbol{B}) \qquad \text{(Eq. 5.9)}$$

to get

$$\frac{\partial S_i^n}{\partial X_k^n} = \begin{cases} \boldsymbol{n}_i^n\times(X_{Bi}^n - X_{Ci}^n) & X_k^n = X_{Ai}^n \\ \boldsymbol{n}_i^n\times(X_{Ci}^n - X_{Ai}^n) & X_k^n = X_{Bi}^n \\ \boldsymbol{n}_i^n\times(X_{Ai}^n - X_{Bi}^n) & X_k^n = X_{Ci}^n \\ 0 & else \end{cases} \qquad \text{(Eq. 5.10)}$$

5.3.3 Constraint Function: Preserving the Angle

The constraint of preservation of angles must be included in our objective function in order to enforce the conformality of the flat map with the original surface. Given that there are T triangles in the tessellated surface, we can define $3T$ equality constraints for the angles of the triangulated mesh. The satisfaction of the conformality constraint is accomplished by adding to the objective function a term that prescribes a high cost for violating these equalities. This term must be a function of the difference between the angles in the current configuration of the surface and those in its original configuration.

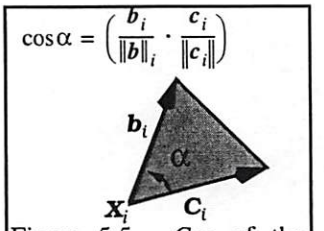


Figure 5.5. *Cos* of the angle between two vectors.

The difference between any one-to-one function of the angles is also equally useful. $Cos\alpha$ is a monotonically decreasing function of α for $0 \le \alpha \le 180$. Therefore, a cost function

defined to minimize the difference between the cosine of the angles of a triangle with respect to some reference will indirectly minimize the difference between the angles of that triangle and those of the reference one. The advantage of the cosine over other monotonic functions of an angle is that it can be easily calculated as the inner product of the unit vectors in the directions parallel to the sides of angle (Figure (5.5)).

Based on this argument we have defined the angular constraint function I_a as

$$I_a = \frac{1}{3T}\sum_{i=0}^{3T-1}(Cos\alpha_i^n - Cos\alpha_i^0)^2 \qquad \text{(Eq. 5.11)}$$

where T is the total number of triangles in the mesh and n is the iteration number starting from zero.

The dependency of I_a on the position of the vertices of the surface becomes clear when $Cos\alpha_i$ is replaced with its equivalent in terms of the sides of the angle α_i.

$$I_a = \frac{1}{3T}\sum_{i=0}^{3T-1}\left[\left(\frac{\boldsymbol{b}_i^n}{\|\boldsymbol{b}_i^n\|}\cdot\frac{\boldsymbol{c}_i^n}{\|\boldsymbol{c}_i^n\|}\right) - Cos\alpha_i^0\right]^2 \qquad \text{(Eq. 5.12)}$$

Using $\boldsymbol{b}_i = \boldsymbol{X}_{Ci} - \boldsymbol{X}_{Ai}$ and $\boldsymbol{c}_i = \boldsymbol{X}_{Bi} - \boldsymbol{X}_{Ai}$ we will have:

$$I_a = \frac{1}{3T}\sum_{i=0}^{3T-1}\left[\left(\frac{\boldsymbol{X}_{Ci}^n - \boldsymbol{X}_{Ai}^n}{\|\boldsymbol{X}_{Ci}^n - \boldsymbol{X}_{Ai}^n\|}\cdot\frac{\boldsymbol{X}_{Bi}^n - \boldsymbol{X}_{Ai}^n}{\|\boldsymbol{X}_{Bi}^n - \boldsymbol{X}_{Ai}^n\|}\right) - Cos\alpha_i^0\right]^2 \qquad \text{(Eq. 5.13)}$$

In order to calculate ∇I_a we need to determine $\frac{\partial I_a}{\partial \boldsymbol{X}_k}$. Taking partial derivative of I_a with respect to k^{th} vertex we will get:

$$\frac{\partial I_a}{\partial \boldsymbol{X}_k^n} = \frac{2}{3T}\sum_{i=0}^{3T-1}(\cos\alpha_i^n - \cos\alpha_i^0)\frac{\partial}{\partial \boldsymbol{X}_k^n}\cos\alpha_i^n \qquad \text{(Eq. 5.14)}$$

where $\frac{\partial}{\partial \boldsymbol{X}_k^n} = \left(\frac{\partial}{\partial X_{kx}^n}, \frac{\partial}{\partial X_{ky}^n}, \frac{\partial}{\partial X_{kz}^n}\right)^t$ and

$$\frac{\partial}{\partial X_k}\cos\alpha_i^n = \begin{cases} -\dfrac{(\boldsymbol{b}+\boldsymbol{c})}{\|\boldsymbol{b}\|\|\boldsymbol{c}\|} + \cos\alpha_i^n\left(\dfrac{\boldsymbol{b}}{\|\boldsymbol{b}\|^2} + \dfrac{\boldsymbol{c}}{\|\boldsymbol{c}\|^2}\right) & X_k^n = X_{Ai}^n \\ \dfrac{\boldsymbol{b}}{\|\boldsymbol{b}\|\|\boldsymbol{c}\|} - \cos\alpha_i^n \dfrac{\boldsymbol{c}}{\|\boldsymbol{c}\|^2} & X_k^n = X_{Bi}^n \\ \dfrac{\boldsymbol{c}}{\|\boldsymbol{c}\|\|\boldsymbol{b}\|} - \cos\alpha_i^n \dfrac{\boldsymbol{b}}{\|\boldsymbol{b}\|^2} & X_k^n = X_{Ci}^n \\ 0 & else \end{cases} \quad \text{(Eq. 5.15)}$$

5.3.4 Perturbation Function: Smoothing the Surface

When formulating our optimization problem we ideally are looking for a global minimum of function I_c subject to the constraints I_s and I_a. However, both from the theoretical and computational points of view, we must in many circumstances be content with a relative minimum point. Theoretically, when deriving necessary conditions based on differential calculus, or practically when searching for the minimum point by a convergent stepwise procedure, comparisons of the values of nearby points is all that is possible. Global solutions can only be found if the problem possesses certain convexity properties that guarantee that any local minimum is a global minimum. In order to be able to take advantage of these convexity properties the objective function and the constraint set must both be convex [2][38], a requirement that is obviously not satisfied for our problem.

Common gradient-search techniques that are often used for solving optimization problems quickly become trapped in local minima. Local minima are found in an iterative gradient based minimization routine in which constraints are enforced softly using penalty functions, in points at which the gradient of the cost plus penalty is zero.

To deal with the unavoidable problem of local minima we have considered a fourth energy functional I_p, in our total objective function that will become effective only when the solution is trapped in a local minimum. This functional is essentially a spring force function that drives the surface to a smoother configuration forcing the current solution away from local minimum.

$$I_p = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N_i - 1} \|X_i - X_j\|^2 \qquad \text{(Eq. 5.16)}$$

In this relation $X_i : i = 0, 1, \ldots, N-1$ represent the vertices of the surface and $X_j : j = 0, 1, \ldots, N_i - 1$ represent the neighbors of X_i. The derivative of I_p with respect to X_k is:

$$\frac{\partial I_p}{\partial X_k} = \frac{2}{N} \left(\sum_{j=0}^{N_k - 1} (X_k - X_j) - \sum_{j=0}^{N_k - 1} (X_j - X_k) \right) = \frac{4}{N} \sum_{j=0}^{N_k - 1} (X_k - X_j) \qquad \text{(Eq. 5.17)}$$

5.3.5 Merit Function: The complete Energy Function

The complete energy function incorporating the objective function of minimizing convexity and the constraint functions of preserving the areas and the angles is given by

$$I = \beta_c I_c + \beta_s I_s + \beta_a I_a \qquad \text{(Eq. 5.18)}$$

where the coefficients β_c, β_s and β_a define the relative importance of unfolding versus the preservation of areas and angles, respectively. By defining $\gamma_s = \beta_s / \beta_c$ and $\gamma_a = \beta_a / \beta_c$ we obtain

$$I(\gamma_s, \gamma_a, X) = I_c(X) + \gamma_s I_s(X) + \gamma_a I_a(X) \qquad \text{(Eq. 5.19)}$$

Here we have explicitly expressed the dependency of the merit function on X, the position vector of the surface, and γ_a and γ_s, respectively the areal and angular penalty coefficients. The procedure for solving the optimization problem given by Equation (5.19) is this: We let $\{\gamma_{sk}\}, \{\gamma_{ak}\}$, $k = 1, 2, \ldots$ be two sequences tending to infinity such that for each k, $\gamma_{sk} \geq 0$, $\gamma_{sk+1} > \gamma_{sk}$ and $\gamma_{ak} \geq 0$, $\gamma_{ak+1} > \gamma_{ak}$. Then, for each k we solve the problem

$$min \quad I(\gamma_{sk}, \gamma_{ak}, X) \qquad \text{(Eq. 5.20)}$$

to obtain a solution point X_k. Any limit point of the sequence $\{X_k\}$ is a solution to our optimization problem.

The actual minimization of $I(\gamma_{sk}, \gamma_{ak}, X)$ is accomplished using a partial conjugate gradient method with line searching based on Armijo's rule [38].

5.4 Solving the Minimization Problem: Conjugate Gradient Method

There are many algorithms designed to solve nonlinear programming problems that belong to the general class of iterative decent algorithms. The simplest and oldest algorithm in this category is the steepest descent method. The method of steepest descent is defined by the iterative algorithm

$$X_{k+1} = X_k - \alpha_k \nabla I(X_k)$$

where α_k is a nonnegative scalar minimizing $I(X_k - \alpha_k \nabla I(X_k))$. In words, from the point X_k we search along the direction of the negative gradient $-\nabla I(X_k)$ to a minimum point on this line; this point is taken to be X_{k+1}.

Steepest descent converges very slowly. As an alternative, conjugate direction methods are characterized by an accelerated convergence rate with little additional computational overhead. The most widely used conjugate direction method is the conjugate gradient method in which the variable X_k is iteratively updated according to

$$X_{k+1} = X_k + \alpha_k d_k \qquad \text{(Eq. 5.21)}$$

where α_k is obtained by a line search

$$I(X_k + \alpha_k d_k) = \min_{\alpha} I(X_k + \alpha d_k) \qquad \text{(Eq. 5.22)}$$

and d_k is generated by

$$d_k = -\nabla I(X_k) + \lambda_k d_{k-1} \qquad \text{(Eq. 5.23)}$$

where

$$\lambda_k = \frac{\nabla I(X_k)^t (\nabla I(X_k) - \nabla I(X_{k-1}))}{\nabla I(X_{k-1})^t \nabla I(X_{k-1})} \qquad \text{(Eq. 5.24)}$$

The conjugate gradient method is designed to give convergence in a finite number of steps for quadratic functions. Since our problem is not quadratic we must deal with the loss of conjugacy that results from non quadratic terms in the cost function. Also, because of inaccurate line searches it is not unusual for the method to start generating inefficient search directions after a few iterations. For these reasons it is important to operate the method in cycles of conjugate direction steps, with the first step in each cycle being a steepest descent step. To decide when we need to restart a cycle we wait until one of the following two conditions is satisfied: either n iterations have taken place since the preceding restart, n being the dimension of the domain space of our minimization problem, or if

$$\left|\nabla I(X_k)^t \nabla I(X_{k-1})\right| > \gamma \left\|\nabla I(X_{k-1})\right\|^2 \qquad \text{(Eq. 5.25)}$$

where γ is a fixed scalar with $0 < \gamma < 1$. This relation is a test on loss of conjugacy, for if the generated directions were conjugate then we would have $\nabla I(X_k)^t \nabla I(X_{k-1}) = 0$

The complete conjugate gradient algorithm using restart is [38]:

Step 1. Given X_0 compute $\nabla I(X_o)$ and set $d_0 = -\nabla I(X_0)$

Step 2. For $k = 0, 1, \ldots, n-1$:

2a. Set $X_{k+1} = X_k + \alpha_k d_k$ where α_k minimizes $I(X_k + \alpha d_k)$.

2b. Compute $\nabla I(X_{k+1})$.

2c. If $k = n-1$ or $\left|\nabla I(X_k)^t \nabla I(X_{k-1})\right| > \gamma \left\|\nabla I(X_{k-1})\right\|^2$ go to *Step 3* else, set

$$d_k = -\nabla I(X_k) + \lambda_k d_{k-1}$$

where

$$\lambda_k = \frac{\nabla I(X_k)^t (\nabla I(X_k) - \nabla I(X_{k-1}))}{\nabla I(X_{k-1})^t \nabla I(X_{k-1})}$$

Step 3. Replace X_0 by X_n and go back to *Step 1.*

An important practical issue relates to the line search accuracy that is necessary for efficient computation. On one hand, an accurate line search is needed to limit the loss of

direction conjugacy, and on the other hand, insisting on a very accurate line search can be computationally expensive. A practical and popular line search method is Armijo's rule [38]. Armijo's rule guarantees that the selected α is neither too large nor too small. To implement Armijo's rule we choose fix scalars s, $\beta \in (0, 1)$ and $\sigma \in (0, 1)$ and we set $\alpha_k = s\beta^{m_i}$ where m_i is the first nonnegative integer m for which the inequality

$$I(X_k) - I(X_k + s\beta^m \boldsymbol{d}_k) \geq -\sigma s\beta^m \nabla I(X_k)' \boldsymbol{d}_k \qquad \text{(Eq. 5.26)}$$

holds. In Equation (5.26) $\boldsymbol{d}_k$ is the descent direction which for the gradient method is equal to the negative gradient $-\nabla I(X_k)$. Usually σ is chosen from the interval $[10^{-5}, 10^{-1}]$ and the reduction factor $\beta \in [0.1, 0.5]$ depending on the accuracy of our estimate of the initial step-size s.

5.5 Results and Discussion

The flattening algorithm introduced in Section (5.3) develops an isometry between any arbitrary surface and a flat plane on the basis of a theorem in differential geometry which states that every equiareal and conformal mapping is isometric [36]. To evaluate the performance of this algorithm we have applied it to two types of data: artificially generated phantoms with well known geometric properties (cylinder, cone, sphere) and real physical data (human cortical surface).

5.5.1 Phantom Studies

Our main focus in the first set of phantom studies was evaluation of the correctness and accuracy of the objective functions in generating conformal and equiareal maps. It is essential to establish the effectiveness of this approach, i.e. using the framework of solving an optimization problem in order to develop a transformation between surfaces, before it can be applied to real data sets. Both developable and non-developable surfaces were considered in this study: a cylinder as the developable surface and a sphere as the non-developable surface.

To evaluate the ability of the energy functionals defined in Section (5.3) to generate optimal maps with minimum areal and angular distortions, we conducted a test in which the functionals were expected to restore the area and angles of triangles forming the tessellated surface of some computer generated phantom that was subjected to distortion. As the distorted surface we used the flat map of the phantom obtained by simple projection of the surface points into a plane. We applied the error minimization procedure as described in Section (5.4), once with the objective of minimizing the areal errors and once for minimization of the angular errors. In the process of minimization, the distribution of distortion was calculated and mapped over the surface of the phantom. In these distortion maps the color assigned to each triangle was chosen to be proportional to the percentage of the relative error corresponding to that triangle. The histogram of areal and angular errors were also calculated as a quantitative measure for the distribution of distortions. In the case of areal distortion for each triangle this error is simply equal to the difference between the area of that triangle in the current configuration and its area in the original three dimensional surface, divided by the average area of the triangles in the three dimensional surface. In other words, the areal error for the ith triangle, e_i^{Area}, is defined as:

$$e_i^{Area} = \frac{A^n{}_i - A^0{}_i}{\frac{1}{T}\sum_{i=0} A^0{}_i} \times 100 \qquad \text{(Eq. 5.27)}$$

where $A^n{}_i$ is the area of the ith triangle after n iterations, $A^0{}_i$ is its area in the original surface and T is the total number of triangles in the tessellation. According to this relation, a negative error indicates a shrunken triangle whereas a positive error corresponds to an expanded one. For angular distortion the error for the ith triangle is defined as:

$$e_i^{Angle} = \frac{1}{3}\sum_{k=1}^{3}\left|\frac{Cos\alpha^n{}_k - Cos\alpha^0{}_k}{Cos\alpha^0{}_k}\right| \qquad \text{(Eq. 5.28)}$$

where $\alpha^n{}_k$, $k = 1, 2, 3$ are the three angles of the ith triangle after n iterations and $\alpha^0{}_k$, $k = 1, 2, 3$ are the same angles in the original 3-D surface. In this case, the error is always non-negative and larger values of error represents more distorted triangles.

A useful indicator for expressing the level of success of an optimization algorithm is the magnitude of the relative error in the quantity that the algorithm is minimizing. The average relative error in area and angle are defined as:

$$\text{Average Relative Error in Area: } E_{Area} = \frac{1}{T}\sum_{i=0}^{T-1} e_i^{Area}$$

$$\text{Average Relative Error in Angle: } E_{Angle} = \frac{1}{T}\sum_{i=0}^{T-1} e_i^{Angle}$$

Ideally, when the objective is completely accomplished the relative error will become zero. We suggest that a relative error in the range of 0.01% to 0.1% is probably satisfactory for most practical purposes.

5.5.1.1 Equiareal and Conformal Maps of Cylinder

Figure (5.6) shows a tessellated half-cylinder phantom that is generated by specifying the location of its vertices analytically. Figure (5.7.a) shows the areal distortion for the projectional map of the half-cylinder and figure (5.7.c) shows the angular distortion for the same map. Figures (5.7.b and 5.7.d) display the histogram of areal and angular distortions for this map, respectively. As can be seen, the distortion is minimum in the center of the map with a symmetrically increasing trend towards the boundaries.

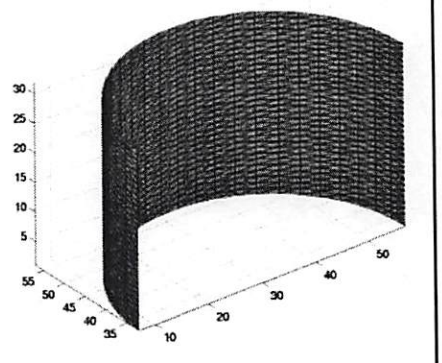


Figure 5.6. Tessellated half-cylinder.

In Figure (5.8) we have demonstrated the change in the distribution of areal distortion over

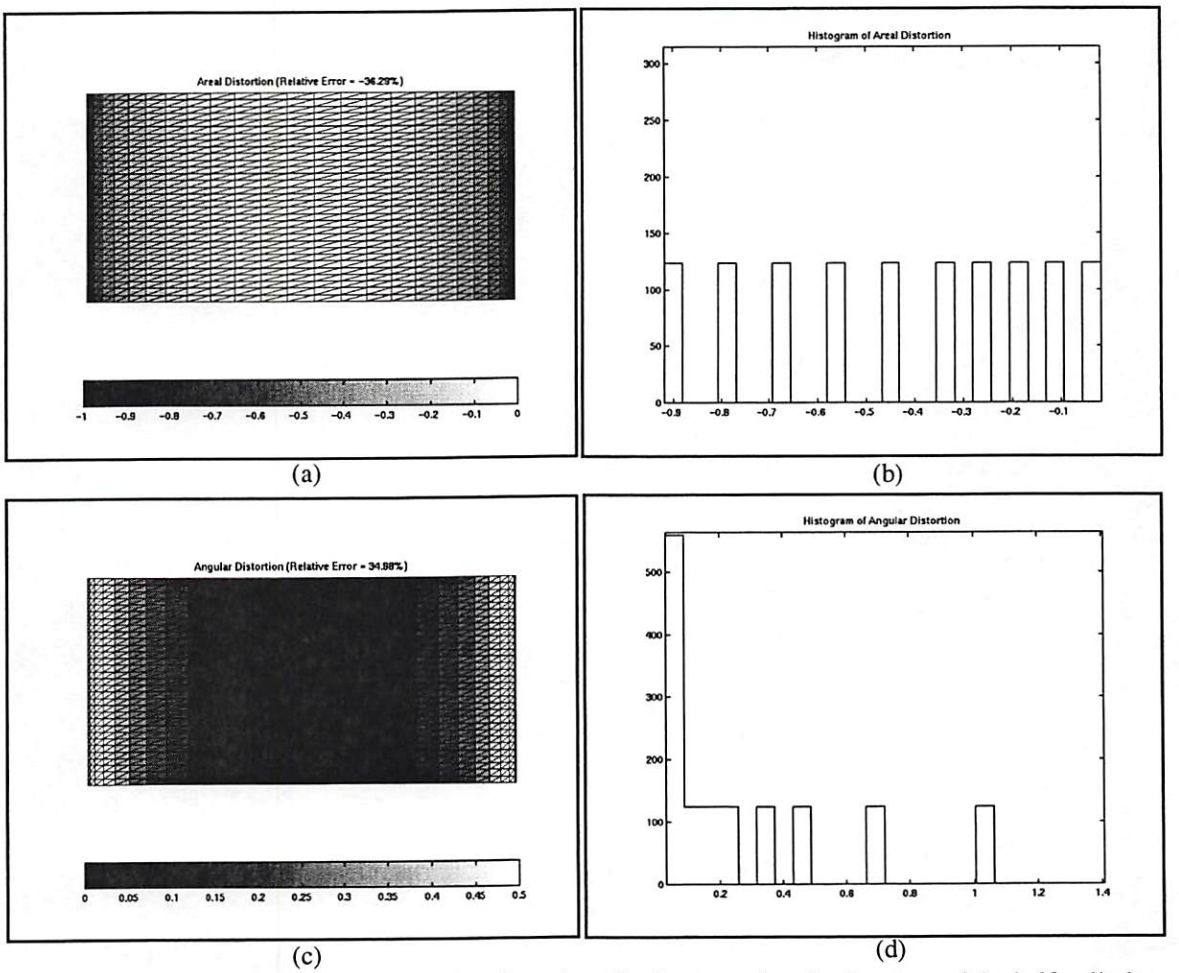


Figure 5.7. Distribution of areal and angular distortions for the normal projection map of the half-cylinder. (a) Areal distortion map. (b) Histogram of areal distortion. (c) Angular distortion map. (d) Histogram of angular distortion.

time as the difference between the area of the flat map and the half-cylinder is minimized by the iterative process of conjugate gradient. The figure clearly shows that the map expands from all four corners in order to accommodate the compressed triangles in the left and right sides. From Figure (5.8) we observe that the iterative process of conjugate gradient not only reduces the total areal error, but also distributes this error approximately uniformly over the entire map. While in the beginning of the process the histogram of areal error is concentrated on the negative side of the axis at the end it is almost symmetrically centered around zero. This can be verified also by checking the

configuration of triangles in the map. When the process starts the map contains compressed triangles in the left and right borders, but it gradually changes such

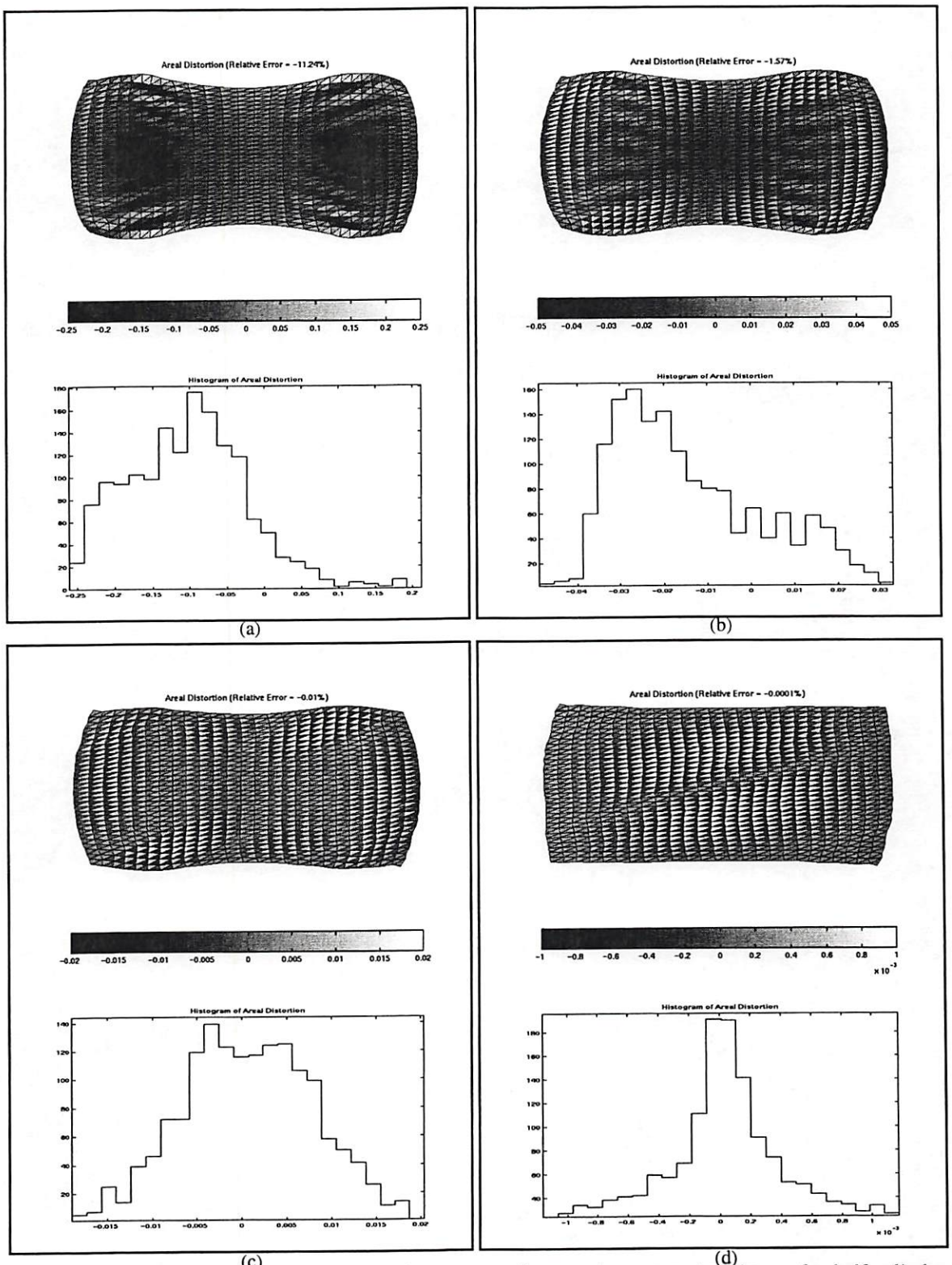


Figure 5.8. Variation of areal distortion in the process of generating an equiareal map for half-cylinder. (a) 10 iterations. (b) 50 iterations. (c) 100 iterations. (d) 10000 iterations.

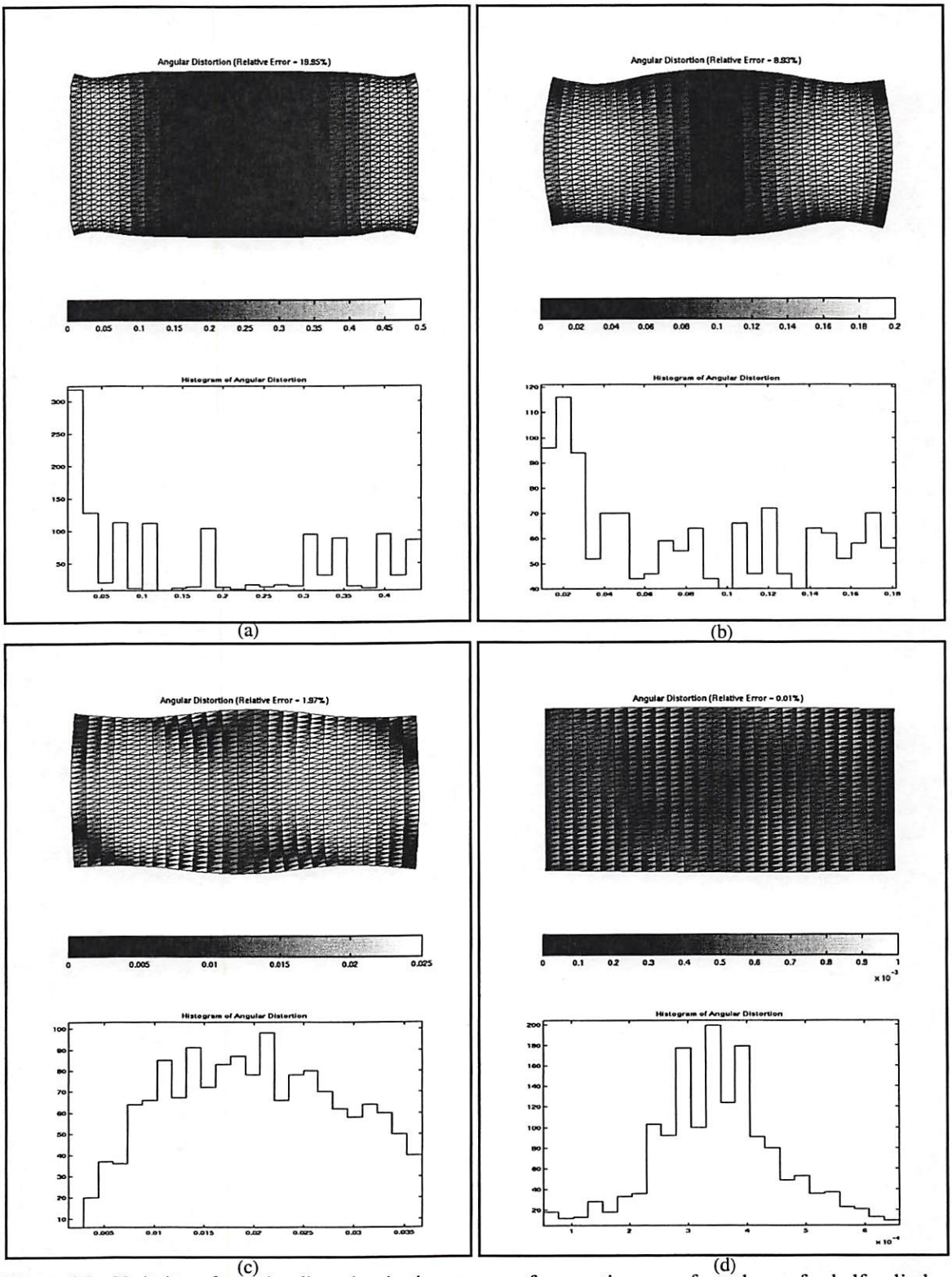


Figure 5.9. Variation of angular distortion in the process of generating a conformal map for half-cylinder. (a) 10 iterations. (b) 500 iterations. (c) 2500 iterations. (d) 10000 iterations.

that at the end, pairs of compressed and expanded triangles (black and white triangles in Figure (5.8.d)) are uniformly spread all over the map. Figure (5.9) shows the angular distortion map of the half-cylinder in different stages of the development of its conformal map. This figure indicates that the reconfiguration of the points is in the direction of aligning the sides of adjacent triangles and making them right triangles. As the sequence of errors converges to zero we would expect that all triangles reshape to a right triangle. As expected, the concentration of errors is on the two extreme sides of the map where the triangles are significantly distorted due to the initial projection into the plane. Again, based on the distortion maps and histograms we can verify that as a result of the iterative minimization process the angular error is reduced and distributed approximately uniformly over the map.

5.5.1.2 Equiareal and Conformal Maps of Hemisphere

Figure (5.10) is a picture of the tessellation of an analytically defined hemisphere. The areal and angular distortion maps of the normal projection of the hemisphere to the plane of its greatest circle, as well as the histogram of areal and angular errors are displayed in Figure (5.11). We can see that the distortion is minimum on the top of the hemisphere where the tangent plane is parallel to the plane of projection and it increases as we move to the boundaries of the surface. We also note that the initial total areal error of -49.97% conforms with the results one would expect from mathematical calculation. Since the area of the greatest circle (πR^2) is half of the total area of the hemisphere ($2\pi R^2$), projecting the hemisphere onto the plane of its greatest circle will naturally result in 50% error.

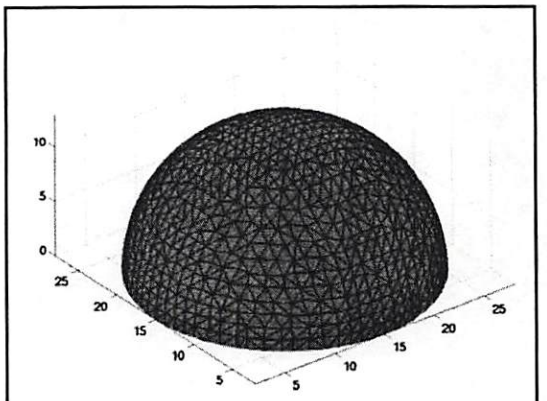


Figure 5.10. Tessellated hemisphere.

Similarly to the half-cylinder, the distribution of areal and angular errors in the process of developing equiareal and conformal maps of the hemisphere were calculated, and a few representative samples of the results are displayed in Figure (5.12) and Figure (5.13), respectively. This series of figures shows that the error is initially concentrated in the regions around the boundaries of the surface, but after a sufficient number of iterations, distributes evenly over the entire map.

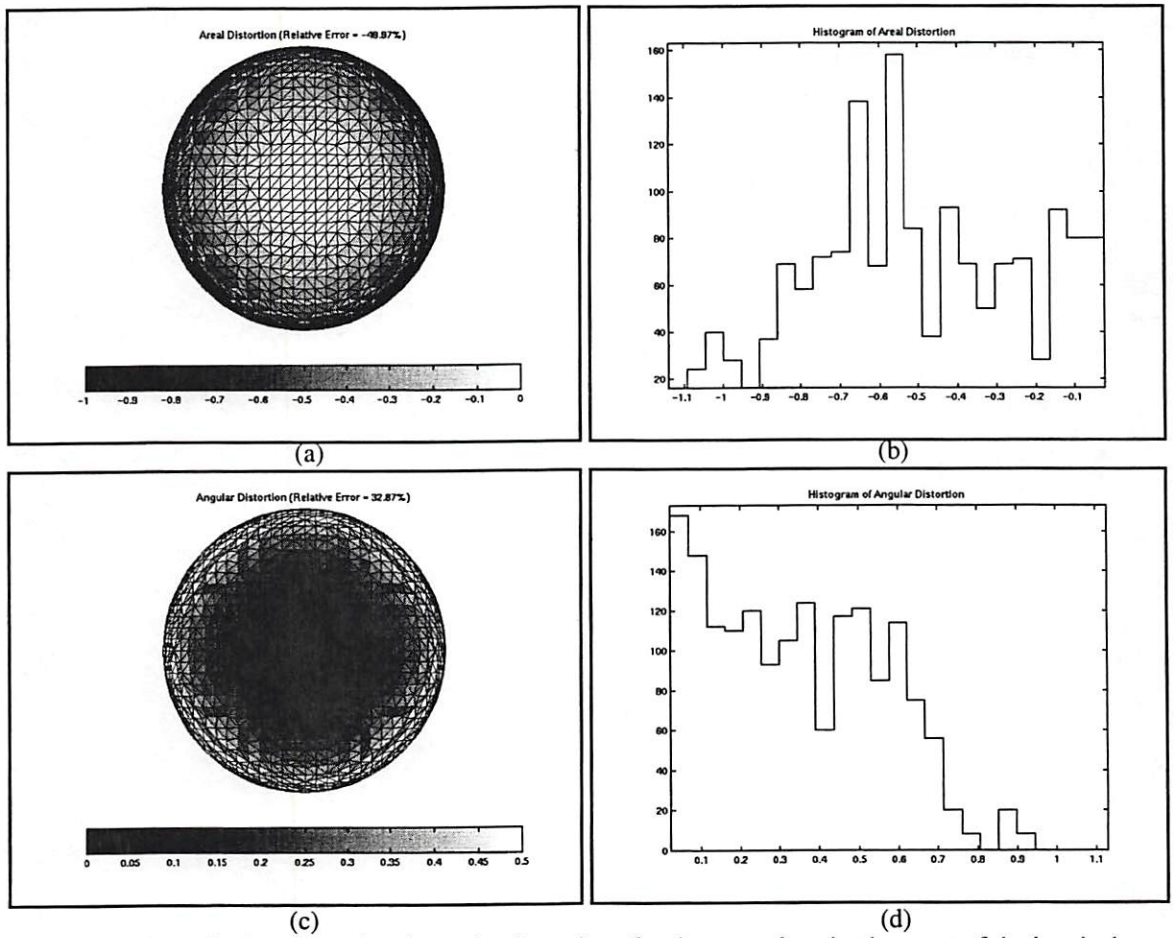


Figure 5.11. Distribution of areal and angular distortions for the normal projection map of the hemisphere. (a) Areal distortion map. (b) Histogram of areal distortion. (c) Angular distortion map. (d) Histogram of angular distortion.

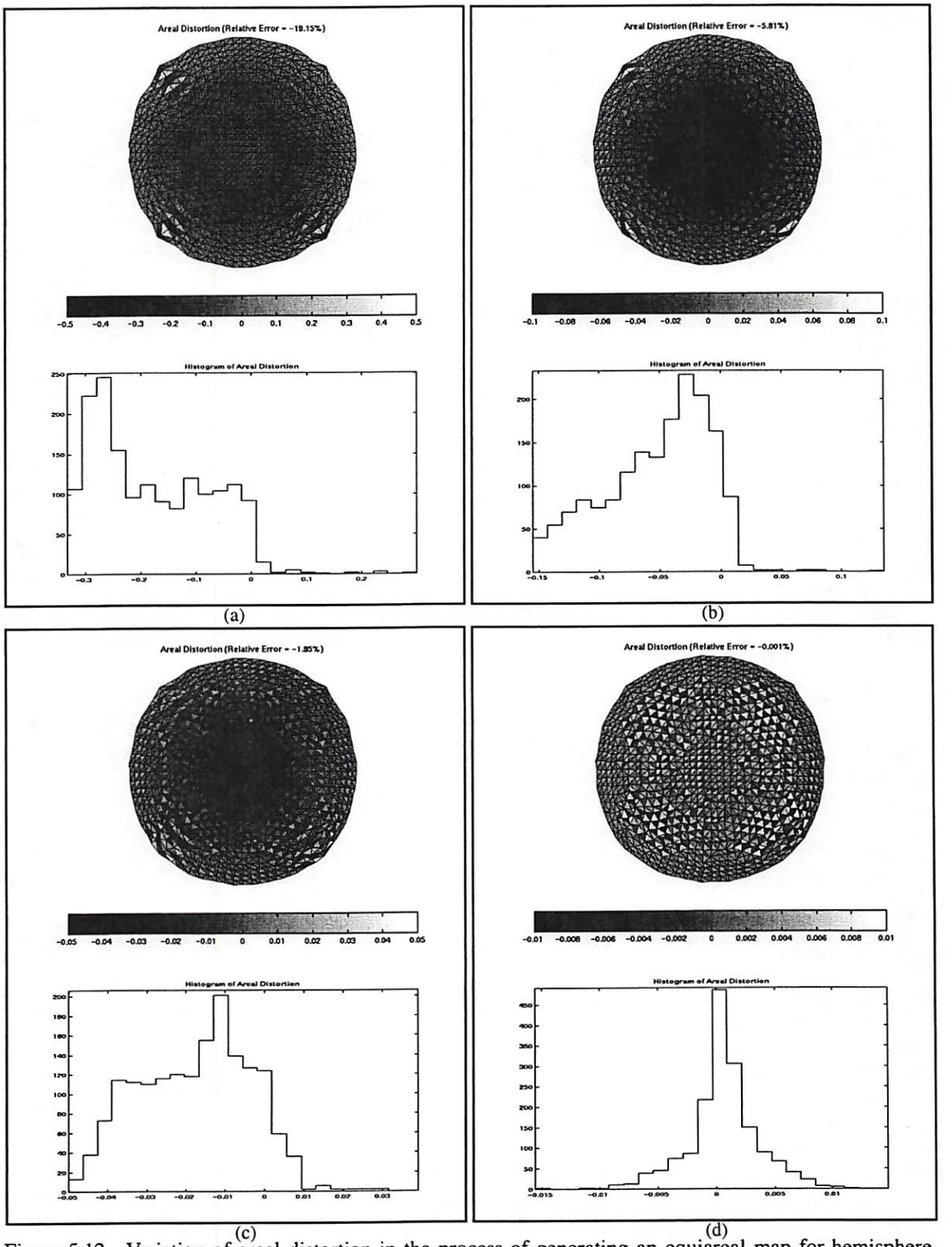


Figure 5.12. Variation of areal distortion in the process of generating an equiareal map for hemisphere. (a) 10 iterations. (b) 50 iterations. (c) 100 iterations. (d) 1000 iterations.

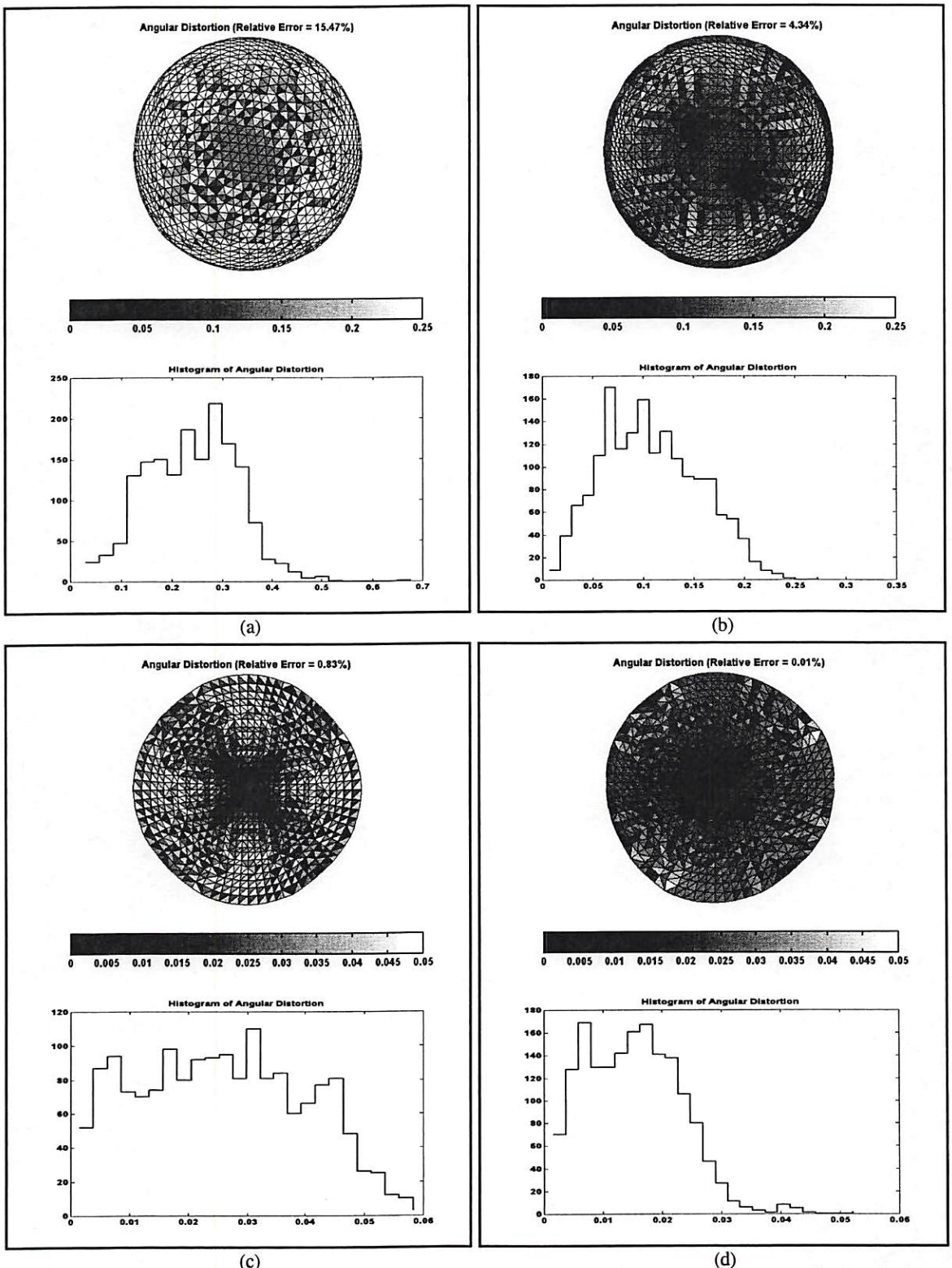


Figure 5.13. Variation of angular distortion in the process of generating an equiareal map for hemisphere. (a) 25 iterations. (b) 150 iterations. (c) 300 iterations. (d) 500 iterations.

5.5.1.3 Isometric Maps of Half-Cylinder and Hemisphere

In the next phase of our phantom studies we wanted to evaluate the accuracy of our method in generating isometric maps of surfaces. By using both constraint functions simultaneously preserving area and angles, the projection map of the phantoms were forced to converge to a simultaneous equiareal and conformal map, or equivalently an isometric map. The combined objective function in this case is

$$I = \beta_s I_s + \beta_a I_a + \beta_p I_p \qquad \text{(Eq. 5.29)}$$

where I_s and I_a represent areal and angular constraint functions respectively, with β_s and β_a determining their relative weights. Since we desire to have both constraints enforced at the same time we set $\beta_s = \beta_a$. As mentioned earlier in Section (5.3.4), this objective function is nonconvex and it is possible that the sequence of solutions for this optimization problem traps in local minima. The last term I_p in Equation (5.29) is the perturbation function which helps the sequence escape from a local minimum. However, since this term effectively smooths the surface through local averaging, it counteracts the other two terms. To reach a compromise between these two opposing effects we use an adaptive mechanism for selecting β_p. Typically we start with $\beta_p = 0$. As soon as the sequence is trapped in a local minimum (this happens when the values of both objective functions I_s and I_a do not change in two consecutive iterations of the conjugate gradient algorithm while the magnitude of the gradient vector is less than a small threshold), we assign a fixed value to β_p (usually $\beta_p = 0.1\beta_s$) for the next iteration of the algorithm. Next time the algorithm gets stuck a fraction (usually 0.1) of the previous β_p will be used and this continues until β_p becomes less than 10^{-5}.

Following the described procedure we applied our algorithm to both phantoms and calculated the final angular and areal distortion maps (Figure (5.14) and Figure (5.15)).

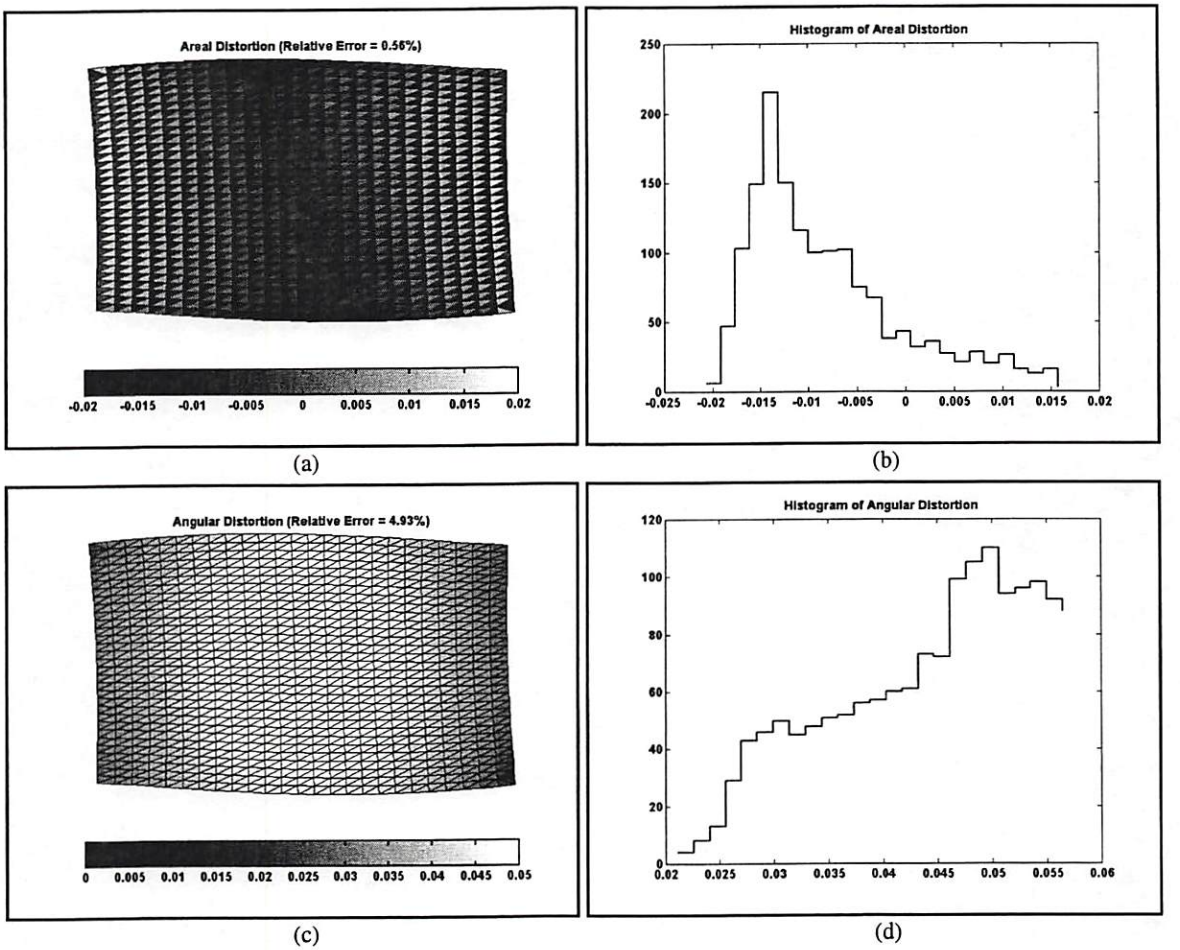


Figure 5.14. Isometric flat map of half-cylinder. (a) Areal distortion map. (b) Histogram of areal error. (c) Angular distortion map. (d) Histogram of angular error.

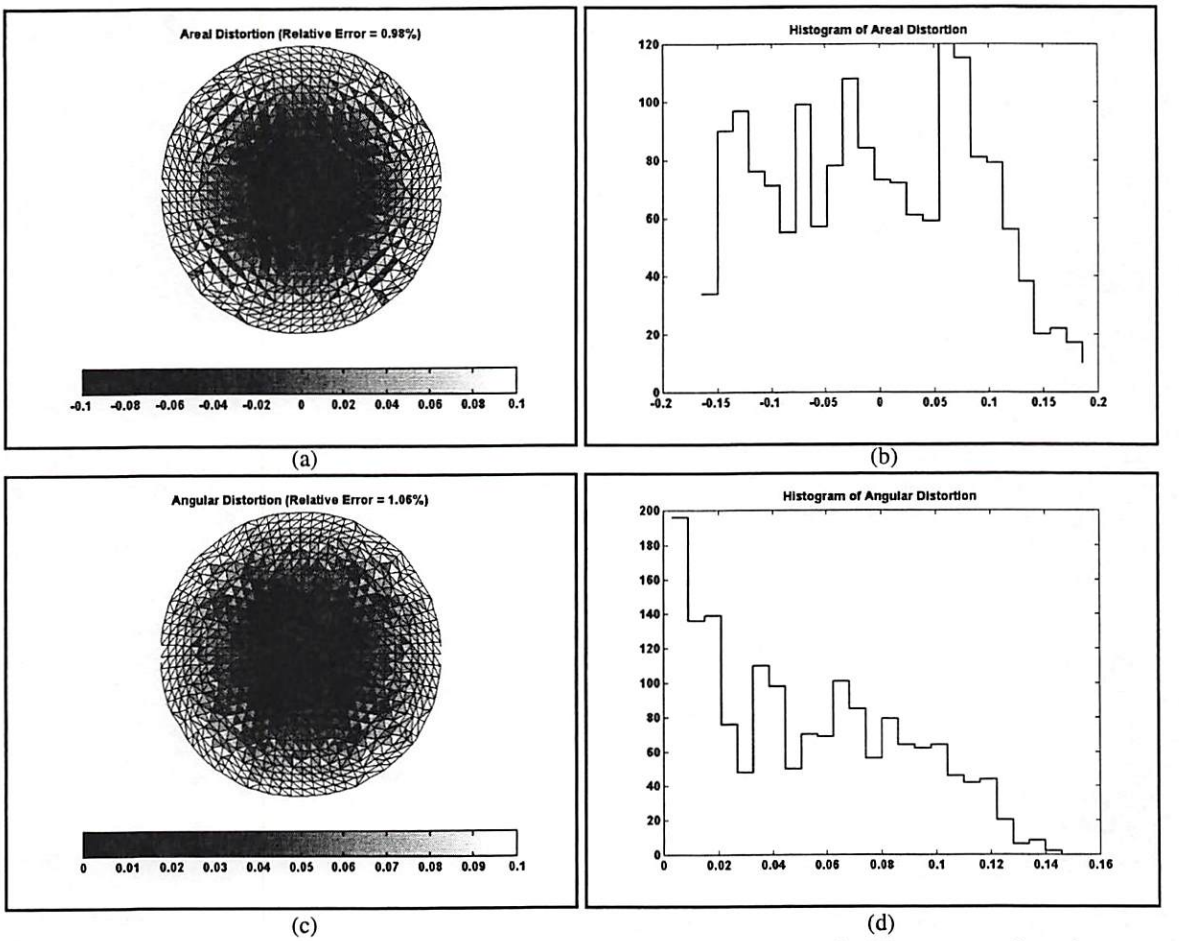


Figure 5.15. Isometric flat map of hemisphere. (a) Areal distortion map. (b) Histogram of areal error. (c) Angular distortion map. (d) Histogram of angular error.

5.5.2 Isometric map of Human Cortical Surface

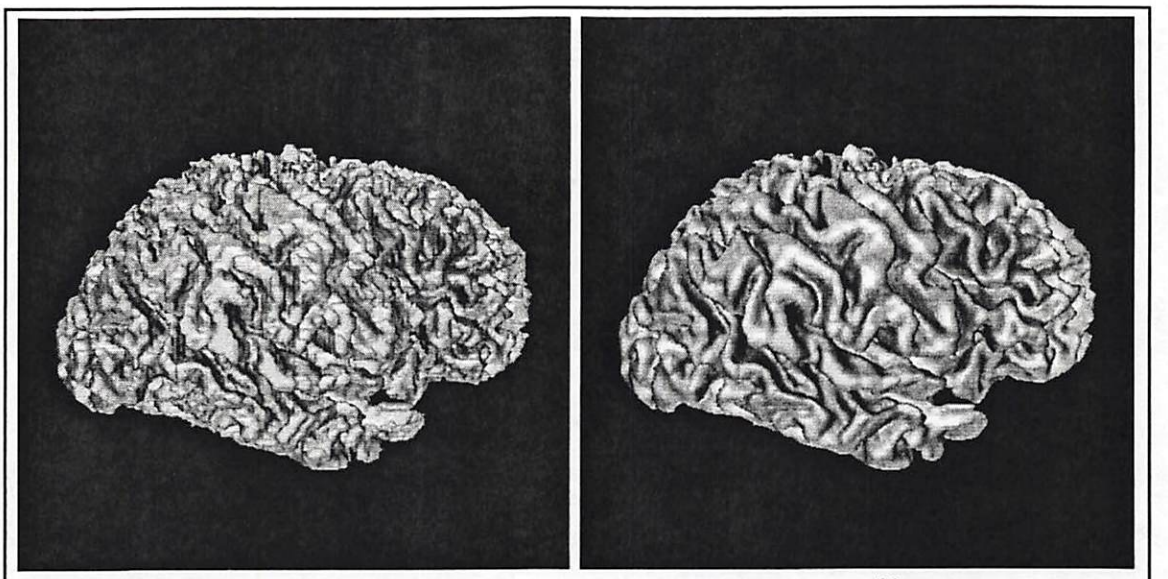
In this section we present the results of applying our algorithm to a human cortical surface. This cortical surface was extracted from the MR images of the head of a normal subject by using a morphological based segmentation method [45][46] and was post processed through manual editing to correct the topology. The corrected volume was then tessellated using the algorithm described in Chapter 2. Since the triangles created by the tesselation process are formed by linking pairs of neighboring points in 3-D, and the neighborhood system used for this purpose only allows a limited number of configurations for the neighboring points, at the voxel scale the tesselation tends to be jagged with a uniform shape pattern. To alleviate this effect, we smooth the initial tessellation using a relaxation operator [23]. Vertices are iteratively repositioned according to

$$X_i^{t+1} = (1-\lambda)X_i^t + \lambda \bar{X}_i \qquad \text{(Eq. 5.30)}$$

where X_i is the position of the vertex i, t is the iteration number and λ is a smoothing parameter in the [0,1] interval (usually $\lambda = 0.75$). X_i is the weighted average of the centers of all triangles that include vertex X_i and is calculated as

$$X_i^t = \frac{1}{A_i^t} \sum_{j \in \mathcal{N}_i} a_j C_j^t \qquad \text{(Eq. 5.31)}$$

where A_i^t is the total area of all triangles that contain vertex X_i, $\mathcal{N}_i$ is the set of all such triangles, a_j is the area of triangle j and C_j is the center of triangle j. The degree of smoothing applied should be sufficient to eliminate all the obvious irregularities, but not so much as to seriously distort the cortical folds. Figure (5.16) shows a sample cortical surface before and after applying the smoothing relaxation operator.



(a) (b)

Figure 5.16. (a) Tessellated cortical surface. (b) Same surface after applying smoothing relaxation

Inflating the topologically corrected cortical surface through minimization of its convexity eventually results in a closed convex surface. In this experiment we were interested in obtaining a planar flat map of the cortex, so rather than the closed surface of the entire brain we used the open surface of a half brain. Being a folded surface, cutting the cortex with a blade parallel to one of the coordinate planes, even if it passes through the inter-hemispheric fissure, does not necessarily generate two open cortical hemispheres. To find the proper cutting angle we used the semi-inflated brain (Figure (5.17.a)) and cut it with a plane with variable orientation passing through a point located on the interhemispheric fissure. The orientation that resulted in two open halves of the inflated brain was used to cut the original cortex. The front and back views of the right cortical hemisphere obtained in this way are displayed in Figure (5.17).(b-c).

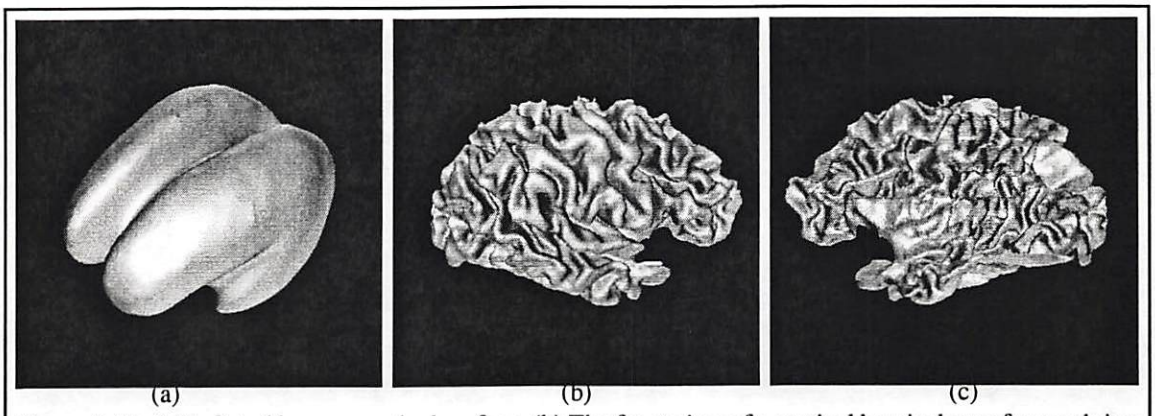


Figure 5.17. (a) Inflated human cortical surface. (b) The front view of a cortical hemisphere after applying a few iterations of relaxation operation to the cortex. (c) The back view of the same cortical hemisphere.

The merit function used for developing the isometric flat map of the cortical hemisphere consists of four components.

$$I = \beta_c I_c + \beta_s I_s + \beta_a I_a + \beta_p I_p \qquad \text{(Eq. 5.32)}$$

The convexity cost function I_c needs to be included in this case because unlike the cylinder or sphere, the cortex is not a convex surface. The minimization routine starts with a relatively large value of β_c ($\beta_c = 10^5 \beta_s$ and $\beta_s = \beta_a$) and with every decade of decrease in the convexity β_c is decreased by a factor of 10. As the surface gradually unfolds, its total convexity continuously decreases. Once the total convexity falls below a pre-defined threshold, we assume the surface has been unfolded sufficiently. After that, β_c is kept constant. At this stage, due to the small relative weights of area preserving and angle preserving constraints during the unfolding process, a significant amount of distortion has been introduced to the surface. This distortion can be reduced by trying to minimize the difference between the angles and area of the triangles in their current configuration and their configuration in the original surface through increasing the relative values of β_s and β_a with respect to β_c. Increasing β_s and β_a has to be done gradually and in steps. An abrupt increase in the weights of any of the two constraints will move the surface in the solution space so rapidly that it may exhibit unstable behavior. As with the

half-cylinder and the hemisphere we use an adaptively changing β_p as described in Section (5.5.1.3), to apply a small perturbation to the surface when the algorithm traps in local minima. Figure (5.18) through Figure (5.24) displays samples of the output generated by the algorithm during the course of development of the flat map of the cortical hemisphere. In this series of pictures, in each figure, the left column in the top row shows the convexity map projected on the current configuration of the surface. The left column in the second and third rows show the areal and angular distortion maps projected on the final flat map of the surface. The right column shows the histogram of convexity, areal error and angular error, respectively from top to the bottom.

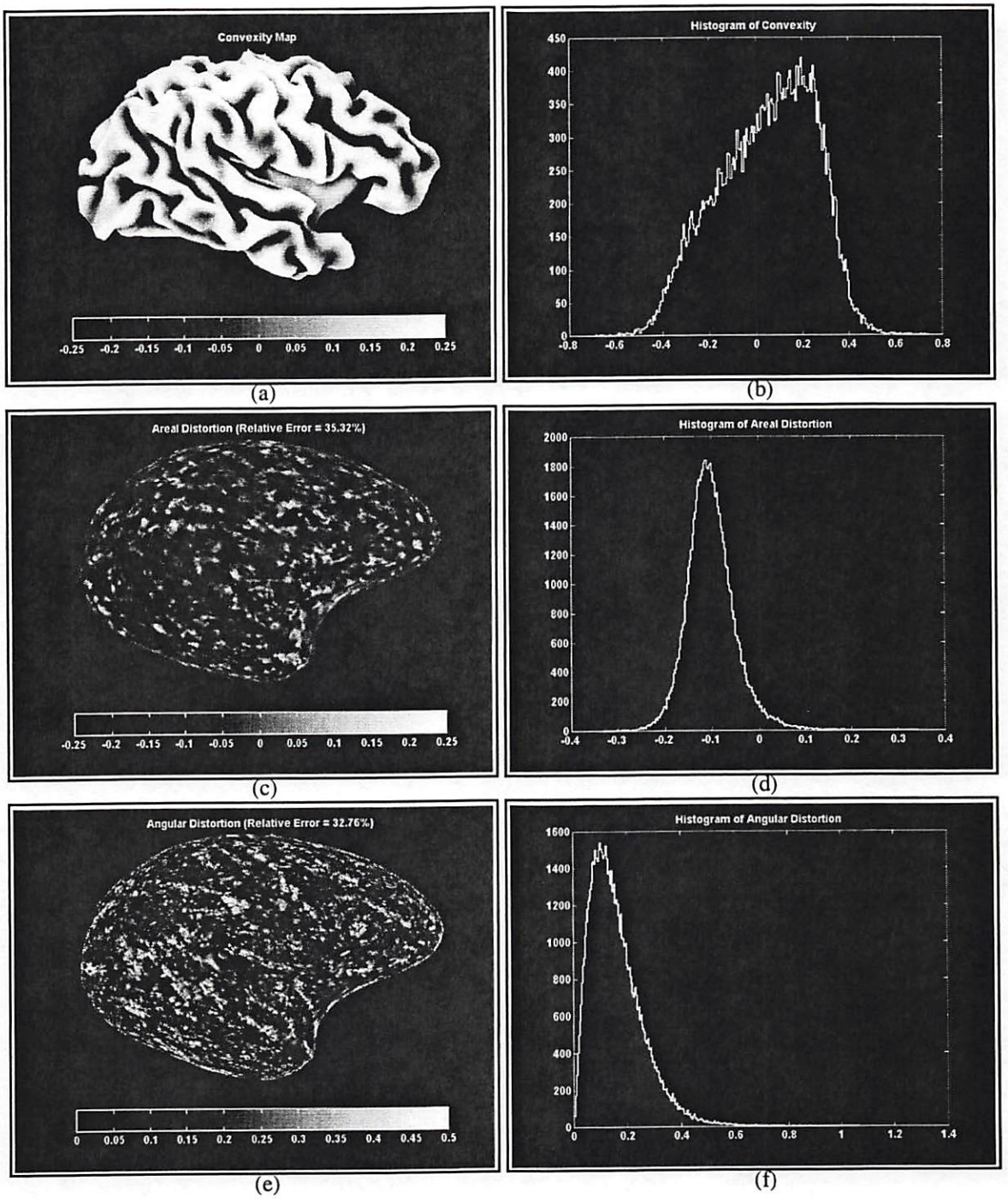


Figure 5.18. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 250 iterations with $\beta_c = 10^5$, $\beta_s = 1$ and $\beta_a = 1$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

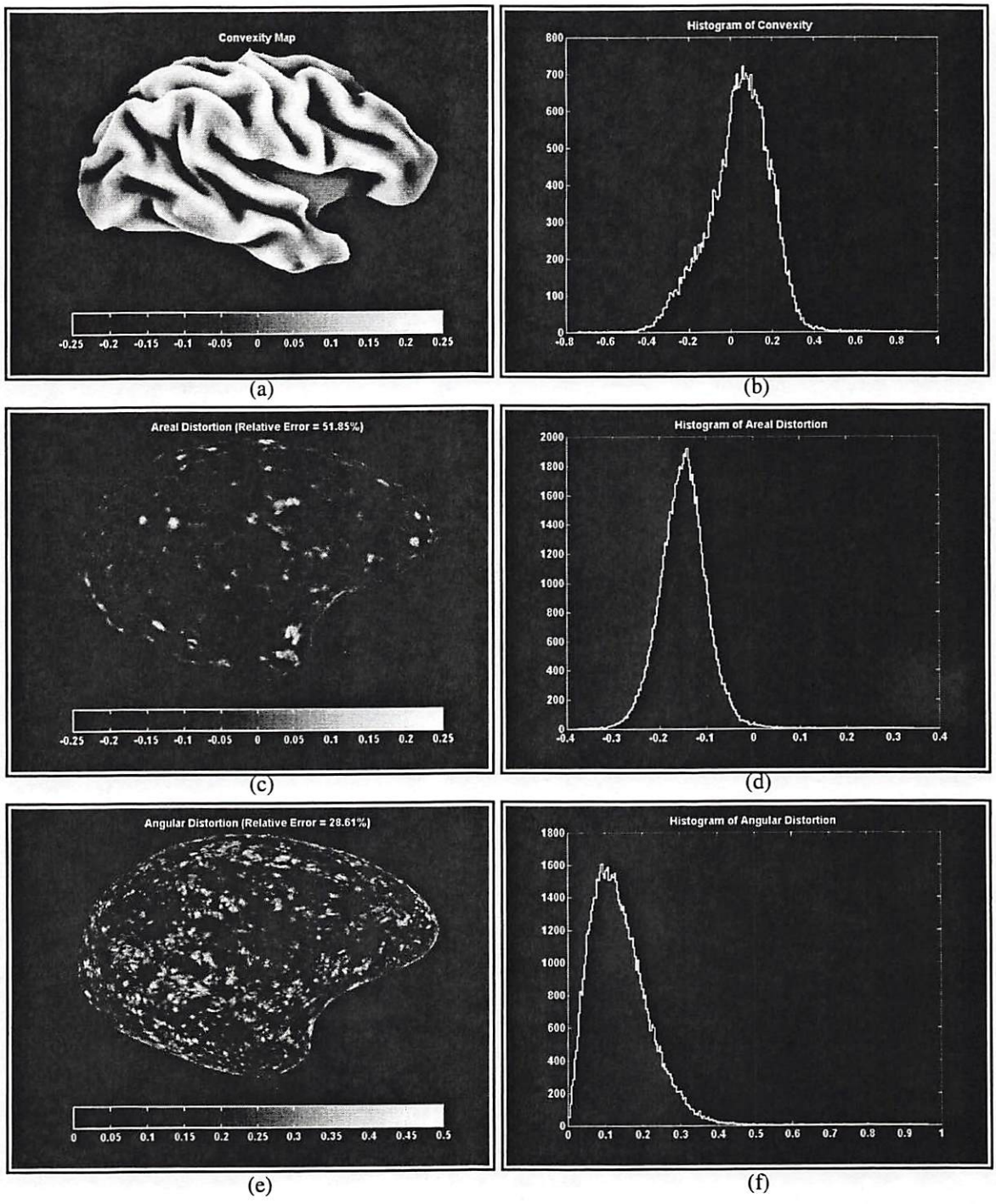


Figure 5.19. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 500 iterations with $\beta_c = 10^4$, $\beta_s = 1$ and $\beta_a = 1$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

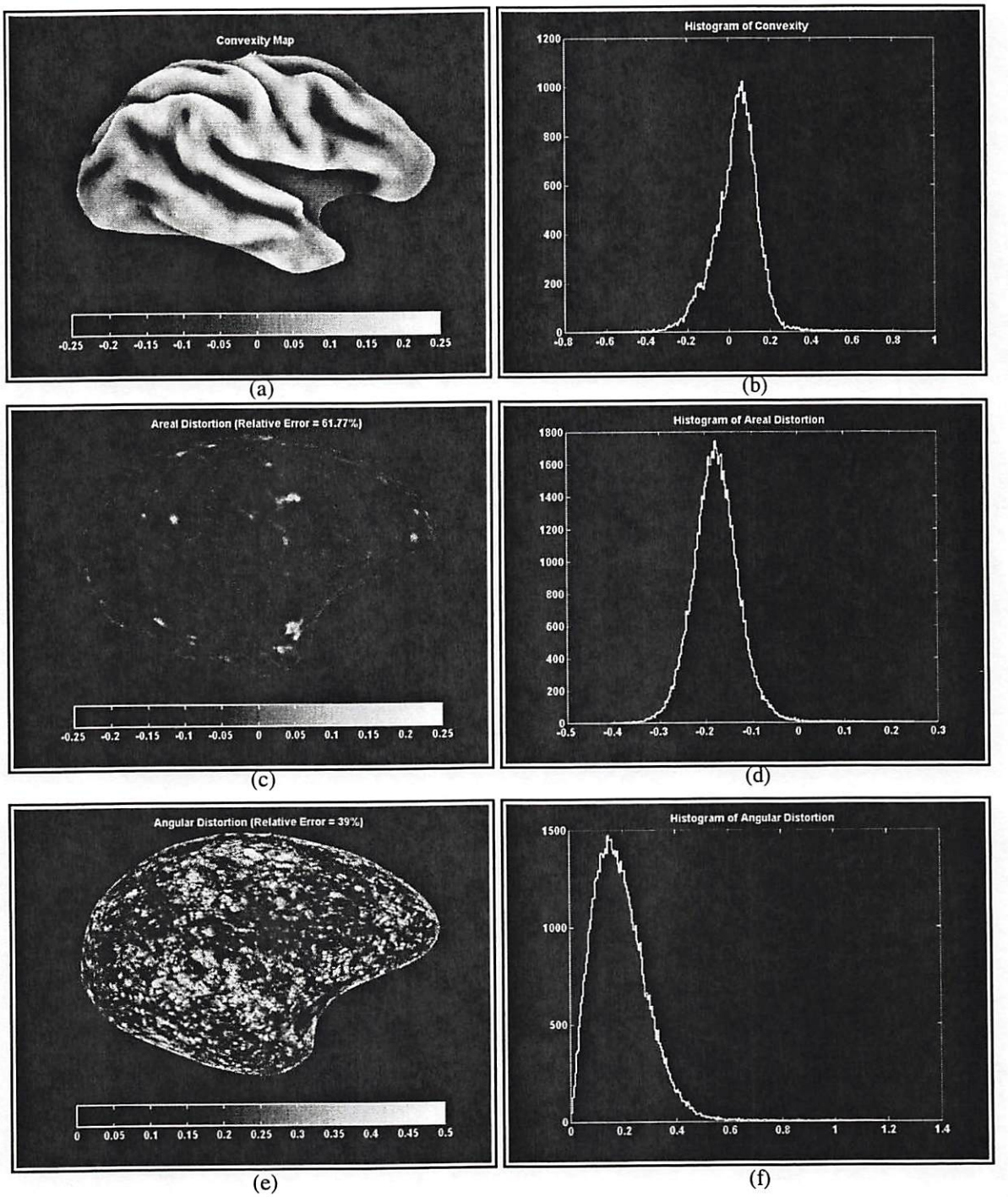


Figure 5.20. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 1000 iterations with $\beta_c = 10^3$, $\beta_s = 1$ and $\beta_a = 1$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

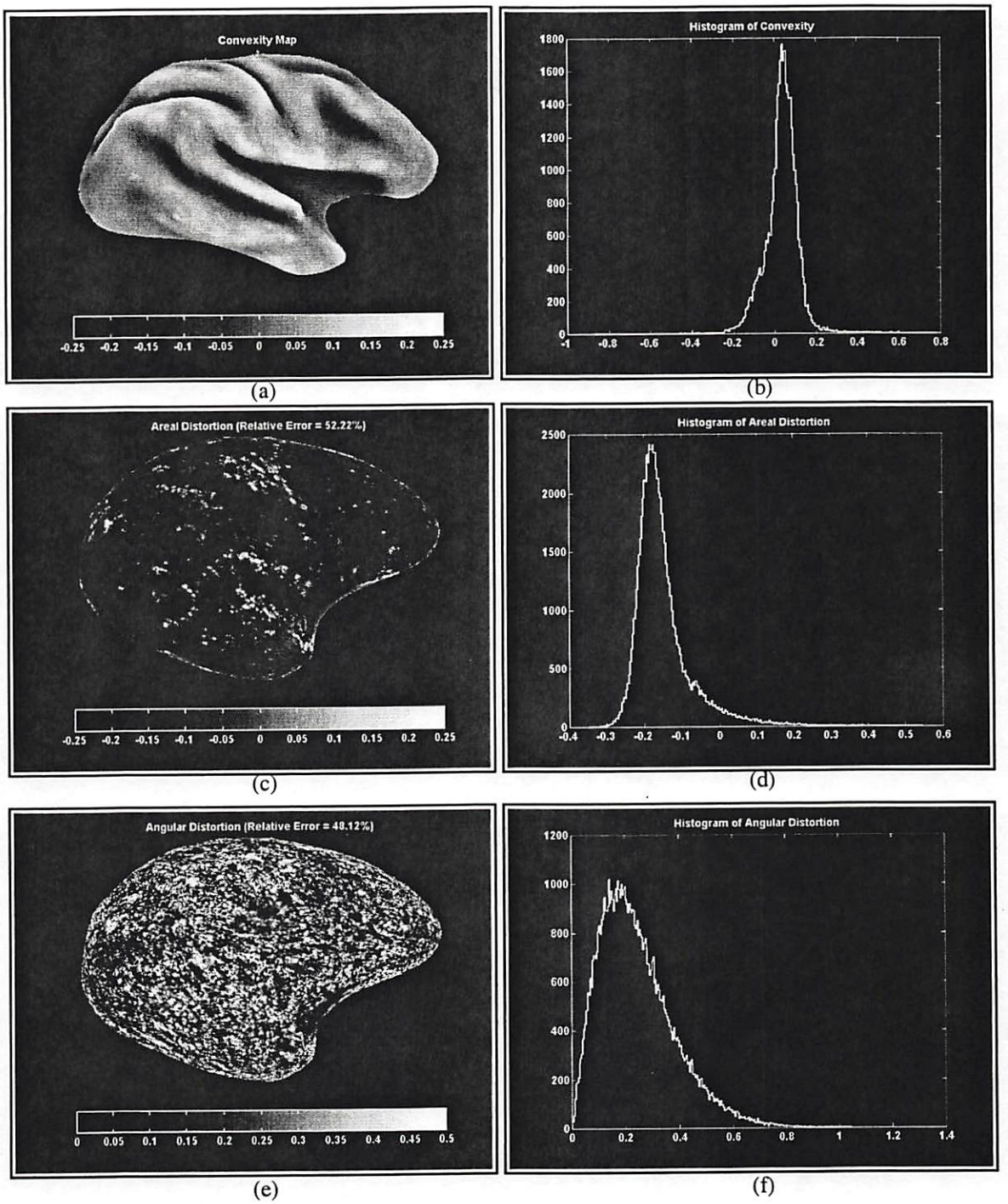


Figure 5.21. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 2500 iterations with $\beta_c = 100$, $\beta_s = 1$ and $\beta_a = 1$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

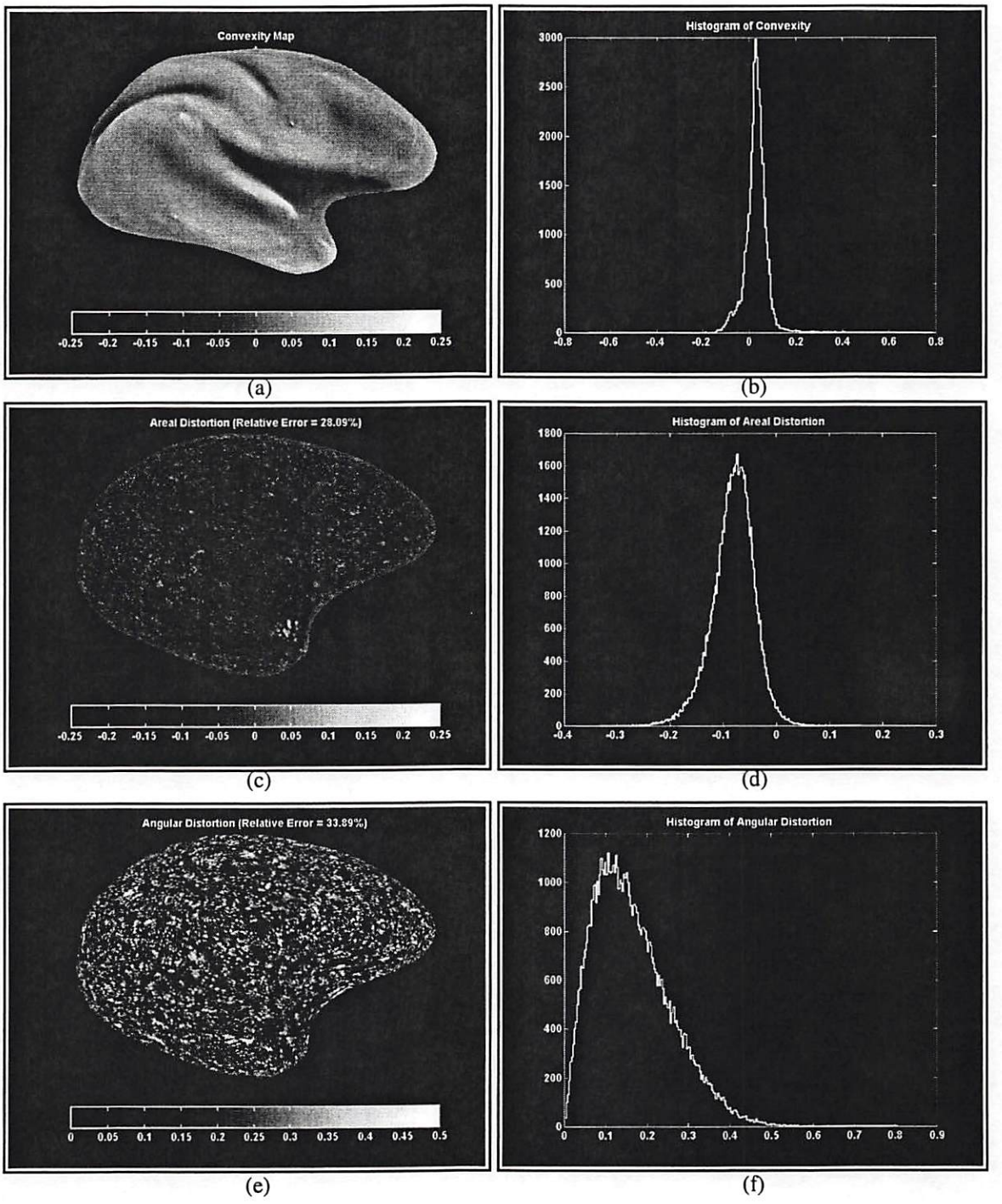


Figure 5.22. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 3000 iterations with $\beta_c = 10$, $\beta_s = 10$ and $\beta_a = 10$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

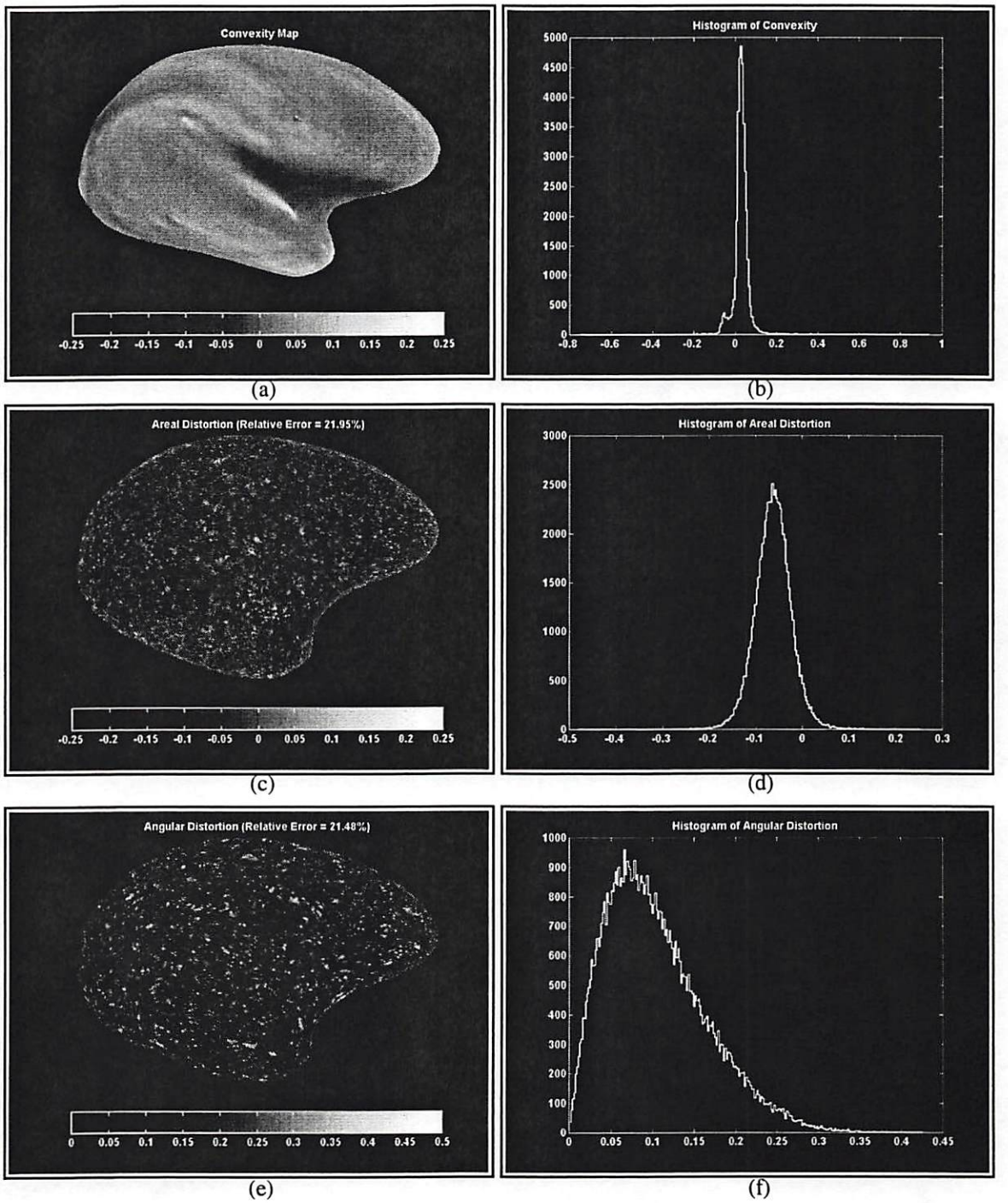


Figure 5.23. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 3500 iterations with $\beta_c = 10$, $\beta_s = 100$ and $\beta_a = 100$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

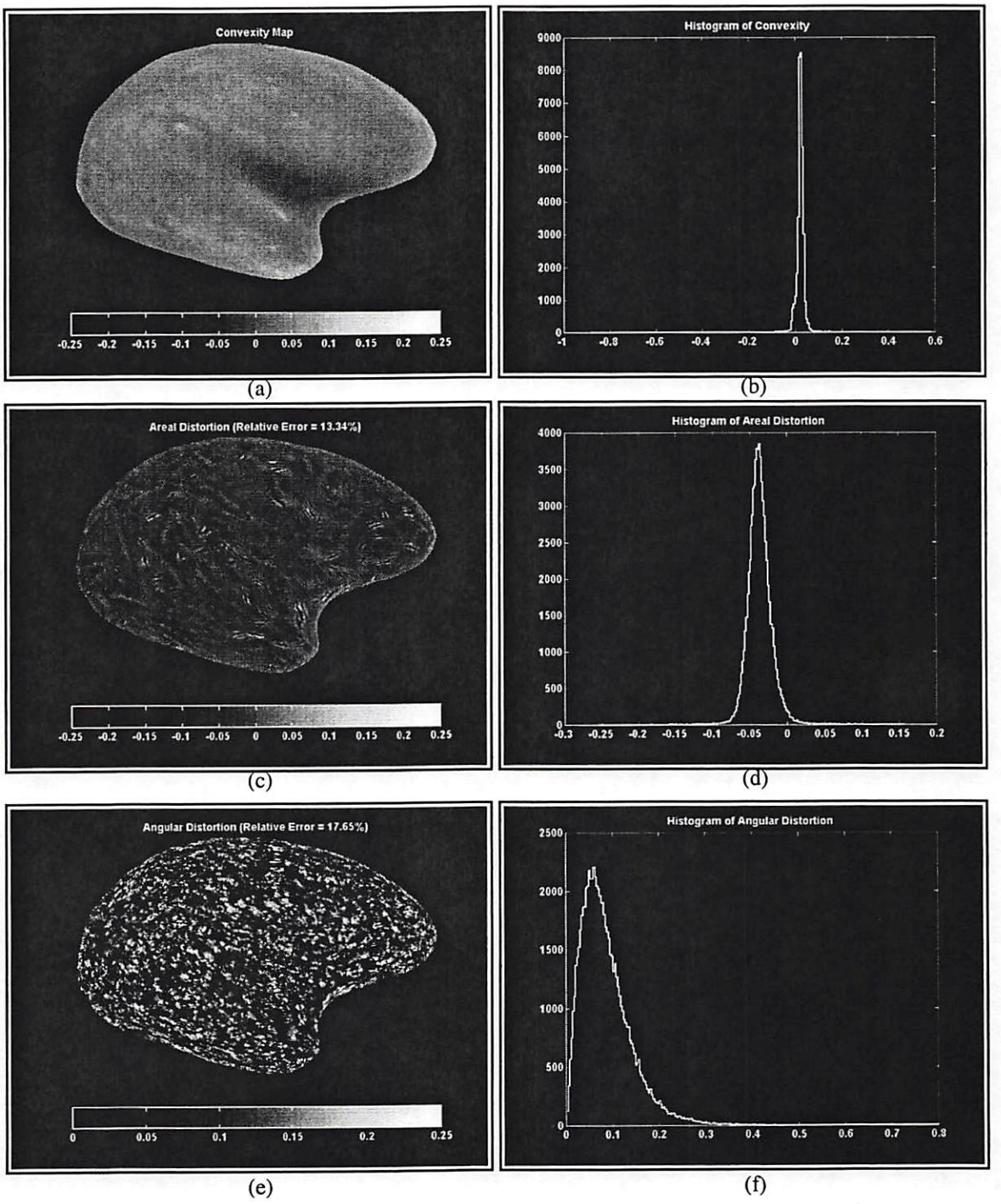


Figure 5.24. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 5000 iterations with $\beta_c = 1$, $\beta_s = 100$ and $\beta_a = 100$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

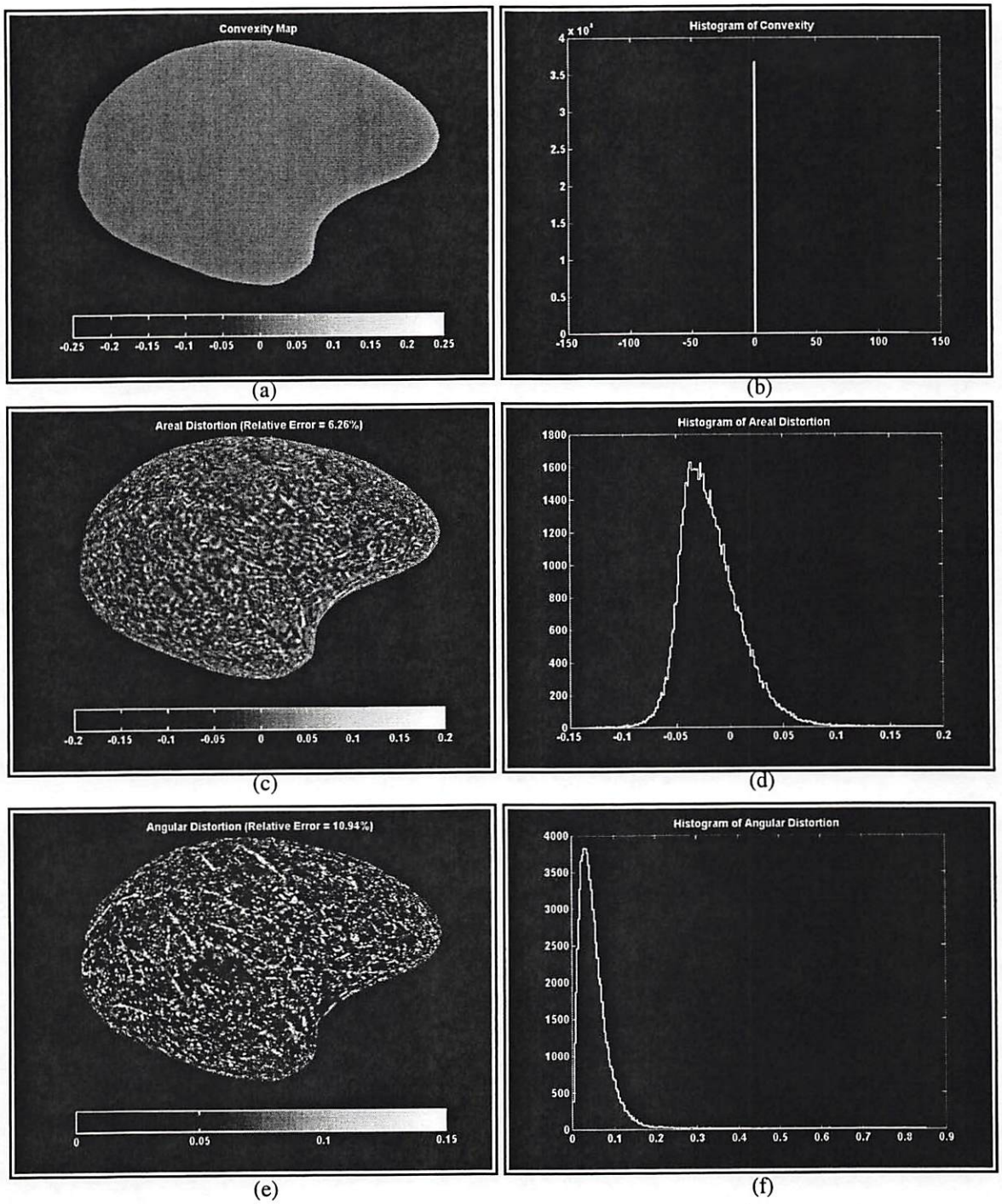


Figure 5.25. Distribution of convexity, areal and angular distortions of a typical right hemispherical human cortex after 10000 iterations with $\beta_c = 0$, $\beta_s = 1000$ and $\beta_a = 1000$. (a) Convexity superimposed over partially unfolded cortex. (b) Histogram of convexity. (c) Areal distortion superimposed over the flat map of cortex. (d) Histogram of areal distortion. (e) Angular distortion superimposed over the flat map of cortex. (f) Histogram of angular distortion.

5.6 Application: Multiresolution Representation of Cortical Surface

In many applications, using a multiresolution strategy can save computational time. In a multiresolution framework, global shape changes are first made on a coarse resolution representation of the surface, and then transferred to progressively finer-resolution representations. The goal is to break down the processing of a complex surface into multiple levels with manageable complexities.

To create a multiresolution representation, the surface of interest must be represented by a regularly connected lattice in which all interior nodes have equal number of neighbors. Currently, no general tessellation technique exists that is capable of creating such a lattice. Potentially, one can resample the polyhedral mesh generated by available techniques and thereby identify a set of nodes belonging to the surface tessellation that forms an appropriately regular lattice. Unfortunately, there is not any reliable way to achieve this objective in a single step, i.e., by direct resampling of a highly convoluted 3-D surface. However, such a resampling is straightforward to achieve on a 2-D surface. Using the isometric flat map of the surface one can resample the 2-D surface into a regular grid and then by interpolation determine the corresponding 3-D coordinates for each node in the resampled grid. The advantage of using isometric maps for this purpose, as opposed to other types of flat maps, is that the side lengths and angles of the triangles will be maintained when mapping back the coordinate data from the 2-D domain to the 3-D space. The first step is to generate a regular hexagonal array of nodes that completely covers and extends slightly outside the 2-D map of the surface. To perform the resampling we have to determine the geometric relationship between every single node of the resampling grid and the triangle in the original map that contains this node. Let us assume that the node $\boldsymbol{g_i}$ falls inside the triangle $\Delta \boldsymbol{abc}$. The triangle $\Delta \boldsymbol{abc}$ defines a local *barycentric* coordinate system [26] in which the coordinates of $\boldsymbol{g_i}$ reflect the fractional areas of the three triangles

formed by this point and any two of the three vertices of Δabc (Figure (5.26)). These coordinates are given by

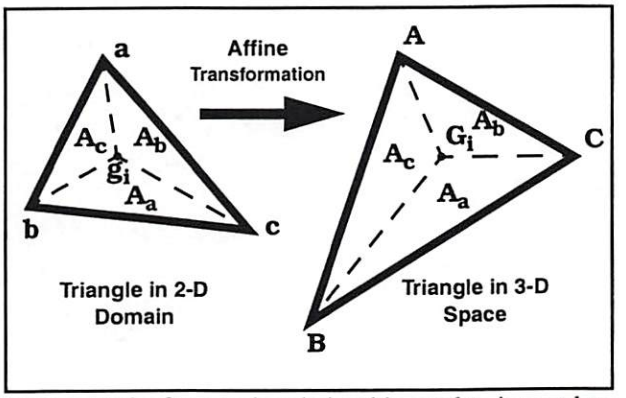


Figure 5.26. Geometric relationship used to interpolate position of nodes during the resampling operation.

$$A_a = \frac{1}{2A_T}[(c - g_i) \times (b - g_i)] \cdot n_i \quad \text{(Eq. 5.33)}$$

$$A_b = \frac{1}{2A_T}[(a - g_i) \times (c - g_i)] \cdot n_i \quad \text{(Eq. 5.34)}$$

$$A_c = \frac{1}{2A_T}[(b - g_i) \times (a - g_i)] \cdot n_i \quad \text{(Eq. 5.35)}$$

where n_i represents the surface normal at point g_i and A_T is the area of Δabc. It can be easily verified that if the areas A_a, A_b and A_c are all positive, then g_i is contained within the triangle Δabc. We use this fact to determine which one of the points on the hexagonal grid are inside the 2-D map. To determine whether each grid point g_i is interior or exterior to the original 2-D surface, the closest surface point S_j to the grid point g_i is identified and Equations (5.33) to (5.35) are used to decide whether g_i is contained within any of the triangles sharing the vertex S_j. For each resampled grid point lying within the original surface, the corresponding location G_i in the 3-D space is determined by a local affine transformation from its location in the 2-D domain. Assuming that the resampled grid point g_i is contained within the triangle Δabc the position of the corresponding node G_i is uniquely specified by the linear combination of the position vectors of the vertices of the corresponding triangle ΔABC with coefficients determined by the barycentric coordinates A_a, A_b and A_c. In other words [26],

$$G_i = A_a A + A_b B + A_c C \quad \text{(Eq. 5.36)}$$

The uniformly resampled reconstruction of the surface can then be downsampled by deleting alternate rows of nodes and alternate nodes within each remaining row to create a lower resolution representation of the surface.

Based on this method we have created a multiresolution representation for the hemispheric phantom and the cortical surface data set whose isometric flat maps are shown in Figure (5.15) and Figure (5.25), respectively. Figure (5.27) displays the original phantom along with the same surface when uniformly resampled in three different resolutions. Figure (5.28) shows the cortical hemisphere at three different resolutions. A magnified picture of a potion of the surface demonstrates the uniformity of the tessellation and the level of detail of surface description.

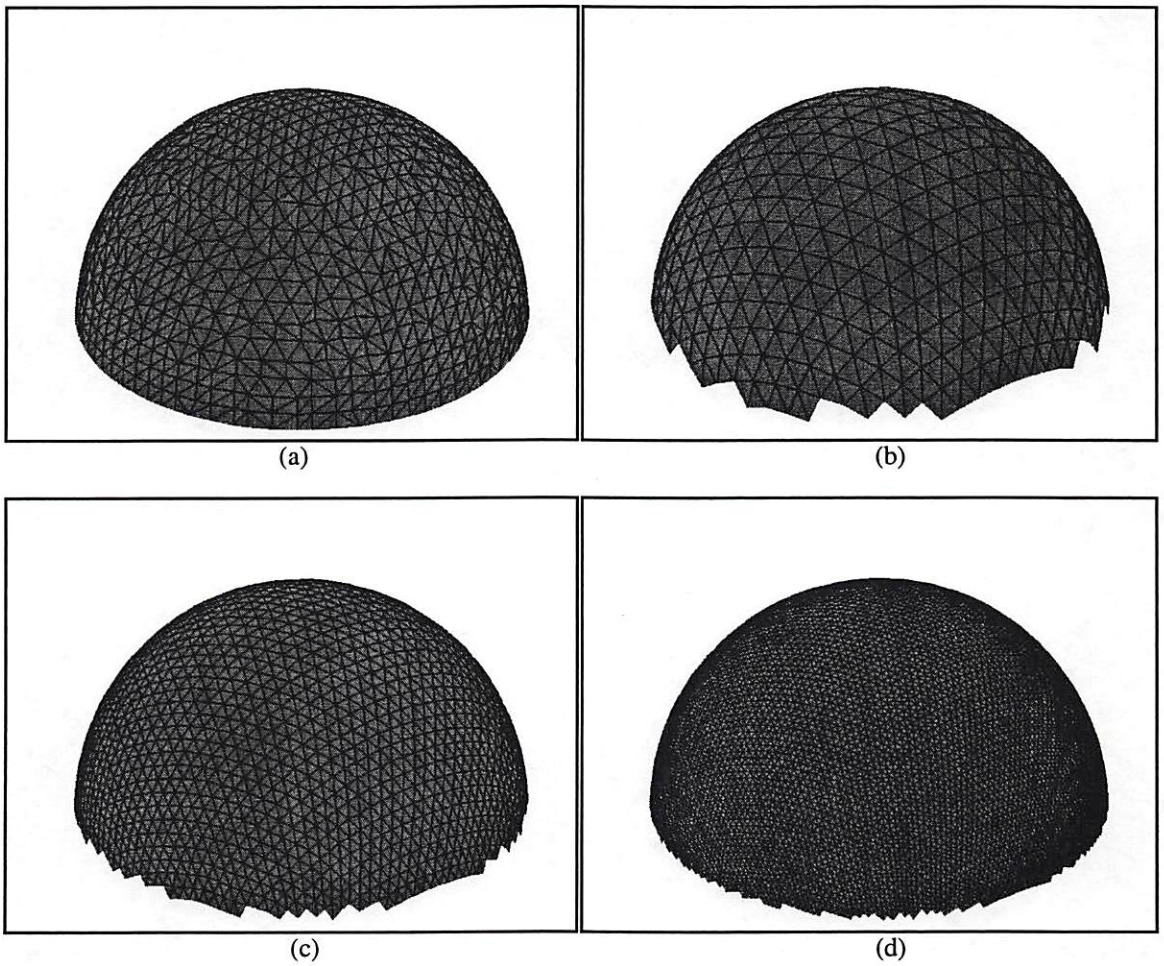


Figure 5.27. Multiresolution representation of a hemisphere created from its isometric flat map.

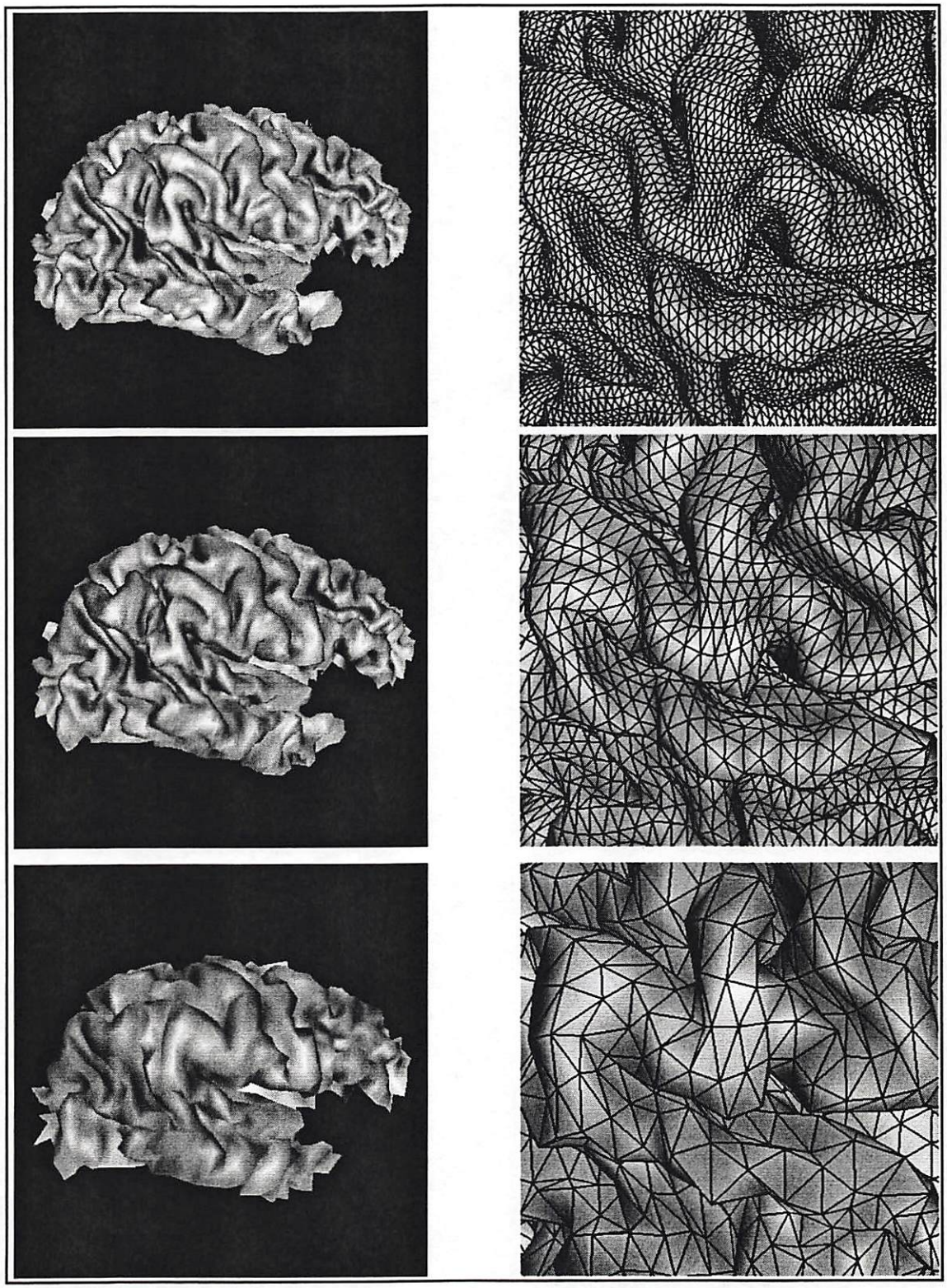


Figure 5.28. Multiresolution representation of a cortical hemisphere created from its isometric flat map.

CHAPTER 6 *Summary and Future Research*

6.1 Conclusion

In this work we have developed a set of tools and methods for geometric modeling and quantitative analysis of complex surfaces. Although in the process of this research we focused on human cortical surface, the developed algorithms and methods are general and equally well applicable to other objects. We presented a method for triangulation of the surface of an object from a set of planar contours representing its constant coordinate slices. Using examples, we showed that this method creates meshes that are comparatively simpler than those generated by the Marching Cubes algorithm. This simplicity, which is practically reflected as a reduction in the number of triangles constituting the mesh, results in savings in processing time for numerical calculations on the surface.

We demonstrated how the polyhedral mesh created by this tool can be used for approximating the surface with local quadratic patches that are appropriate for calculating differential geometric properties of the surface. In particular we showed how minimum, maximum, mean and Gaussian curvatures can be easily calculated from this local description. We used this model to calculate the curvature of a typical human cortical surface and showed that minimum and maximum curvatures can distinctly identify the locations of sulci and gyri on the cortex. We also showed that the mean curvature can provide a true image of the general pattern of cortical foldings. In addition, we calculated the Gaussian curvature of the brain surface and observed that it is a sharply varying

quantity with large variance and mean close to zero. From this we concluded that the human brain surface is not a developable surface but it can be semi-isometrically mapped into a plane.

We also introduced a strategy for developing isometric flat maps of surfaces. Based on the fact that every simultaneous conformal and equiareal mapping is an isometric mapping we formulated a constrained optimization problem consisting of an objective function with a minimum occurring at the unfolded configuration of the surface and two constraint functions guaranteeing the preservation of all areas and angles in the surface. Because of the non-convexity of this model and the possibility of the solution vector becoming stuck in local minima, we have employed a heuristic scheme using a simple smoothing operator that allows us to add a perturbation to the solution vector which facilitates escape from local minima. A conjugate gradient method minimizes the combined energy function and unfolds the surface gradually, deforming it to a flat form corresponding to the global minimum of the total energy function.

We applied this method to data sets representing surfaces of both developable and non-developable types. The result of our experiments confirmed that this method is capable of generating isometric flat maps for developable surfaces and semi-isometric flat maps for non-developable surfaces. We then used this algorithm on some physical data representing the real human brain surface to obtain the semi-isometric flat map of the cortex.

Finally, as a sample application of the developed techniques, we showed how the isometric flat map of a surface can be used for constructing a multiresolution representation for that surface.

All the described methods were tested, first with computer generated phantom data for verifying their correctness, and then with real physical data to demonstrate their practicality and efficiency. All methods provided satisfactory results and hence we assert that they are practically plausible.

6.2 Future Work

There are possibilities for improving the techniques that we have developed and continuing research in areas that we have explored.

Surface tessellation based on coordinate curves seems to be a promising alternative to the Marching Cubes algorithm with the advantage of generating simpler meshes with fewer number of triangles. However, more comprehensive studies are required to evaluate the performance of this method. Issues such as accuracy of the method in reconstructing the detailed shape features of the surface and its complexity compared to other available methods need to be investigated.

An available tool for calculating the curvature of cerebral cortex opens a number of avenues for research. Exploring the correlation between the differential geometry of the cortex and the development of cortical folding is an active area of research where this tool can be fully utilized. Automatic identification of sulcal and gyral organization of the brain is another area that can be explored for possible applications in image registration and matching for inter-subject studies.

The isometric flat map provides many possibilities for effective inter-subject studies. It allows visualization of the distribution of functional data over unfolded maps of the brain. Also it can be considered as a parametric model for the cortical surface; a model appropriate for applying geometrical transformations and apt for surface based registration techniques.

Bibliography

[1] G. Alvarez-Bolado, G. Rosenfeld, and L. W. Swanson (1995), "On Mapping Patterns in The Embryonic Forebrain," *J. Comp. Neurol.* 355, 237-295.

[2] D. P. Bertsekas (1995), *Nonlinear programming*, Athena Scientific, Belmont, Massachusetts.

[3] J. D. Boissonnat (1988), "Shape Reconstruction from Planar Cross-Sections," *Comput. Vision, Graph., Image Proc.* 44, 1, 1- 29.

[4] C. Bohm, T. Greitz, D. Kingsley, B. M. Berggren and L. Olsson (1983), "Adjustable Computerized Brain Atlas for Transmission and Emission Tomography," *Am. J. Neuroradiol.* 4:731:733

[5] S. T. Bok, *The Histonomy of the Cerebral Cortex*, Amesterdam: Elsevier.

[6] M. Bomans, K-H Hohne, U. Tiede and M. Riemer (1990), "3-D Segmentation of MR Images of the Head for 3-D Display," *IEEE Trans. Medical Imaging*, vol. MI-9, pp. 177-183.

[7] Y. Bresler, J. A. Fessler and A. Macovski (1989), "A Bayesian Approach to Reconstruction from Incomplete Projections of Multiple Objects in 3D Domain," *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 8, 840-858.

[8] M. E. Brummer, R. M. Mersereau, R. L. Eisner and R. R. J. Lewine (1993), "Automatic Detection of Brain Contours in MRI Data Sets," *IEEE Trans. Med. Imaging*, vol. 12, 153-166.

[9] G. J. Carman, D. C. Van Essen (1985), "Computational Algorithms for the Production of Two-Dimensional Maps of Cerebral Cortex," *Soc. Neurosci*, Abstr. 11: 1243.

[10] G. J. Carman (1990), *Mappings of the Cerebral Cortex,* Ph.D. dissertation, California Institute of Technology.

[11] G. J. Carman, H. A. Drury and D. C. Van Essen (1995), "Computational Methods for Reconstructing and Unfolding the Cerebral Cortex," *Cerbral Cortex.*

[12] H. N. Christiansen and T. W. Sederberg (1978), "Conversion of Complex Contour Line Definition into Polygonal Element Mosaics," *Comput. Graph.* 12 3, 187-192

[13] H. Cline, W. Lorenson, S. Ludke, C. Crawford and B. Teeter (1988), "Two Algorithms for Three-Dimensional Reconstruction of Tomograms," *Medical Physics*, 15(3):320-327.

[14] L. Collins, C. Holmes, T. M. Peters, A. C. Evans (1996), "Automatic 3-D Segmentation of Neuroanatomical Structures from MRI," *Human Brain Mapping* 3(3), 190-208.

[15] P. N. Cook and S. Batnitsky (1981), "Three-Dimensional Reconstruction from Serial Sections for Medical Applications," *Proc. of the 14th Hawaii Int. Conf. on System Sciences*, 2, pp. 358-389.

[16] P. N. Cook (1981), "Three-Dimensional Reconstruction from Surface Contour for Head C.T. Examinations," *J. Comput. Assist Tomography* 5, 1.

[17] L. T. Cook, P. N. Cook, K. R. Lee, S. Batnitzky, B. Y. S. Wong, S. L. Fritz, J. Ophir, S. J. Dwyer, L. R. Bigongiari and A. W. Templeton (1980), "An Algorithm for Volume Estimation Based on Polyhedral Approximation," *IEEE Trans. Biomed. Eng.* 27, 9, 493-500

[18] A. M. Dale and M. I. Sereno (1993), "Improved Localization of Cortical Activity by Combining EEG and MEG with MRI Cortical Surface Reconstruction: A Linear Approach," *J. Cognitive Neurosci.* 5: 2, pp. 162-176.

[19] C. A. Davatzikos and J. L. Prince (1995), "An Active Contour Model for Mapping the Cortex," *IEEE Trans. Medical Imaging*, Vol. 14, No. 1.

[20] C. A. Davatzikos and R. N. Bryan (1996), "Using a Deformable Surface Model to Obtain a Shape Representation of the Cortex," *IEEE Trans. on Medical Imaging* 15: 785-795.

[21] E. W. Dijkstra (1959), A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik* 1: 269-271.

[22] M. P. Do Carmo (1976), *Differential Geometry of Curves and Surfaces,* New Jersey, Prentice-Hall Inc..

[23] H. A. Drury, D. C. Van Essen, C. H. Anderson, C. W. Lee, T. A. Coogan and J. W. Lewis (1996), "Computerized Mappings of the Cerebral Cortex: A Multiresolution Flattening Method and A Surface-Based Coordinate System," *J. Cognitive Neurosci.* 8: 1. pp. 1-28.

[24] M. J. Durst Letters (April 1998): "Additional Reference to Marching Cubes," *Comput. Graph.*, 22(2):72-73.

[25] A. B. Ekoule, F. C. Peyrin And C. L. Odet (1991), "A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours," *ACM Trans. Graph.* 10, 2, 182-199.

[26] G. Farin (1993), *Curves and Surfaces for Computer Aided Geometric Design*, 3rd ed., San Diego, CA Academic Press.

[27] B. Fischl, M. I. Sereno and A. M. Dale (1999), "Cortical Surface-Based Analysis II: Inflation, Flattening, and a Surface-Based Coordinate System,". *NeuroImage* 9(2):195-207.

[28] H. Fuchs, Z. M. Kedem and S. P. Uselton (1997), "Optimal Surface Reconstruction from Planar Contours," *Commun. ACM* 20, 693-702

[29] S. Ganapathy and T. G. Dennehy (1982), "A New General Triangulation Method for Planar Contours," *Comput. Graph.* 16, 3, 69-75.

[30] L. D. Griffin (1994), "The Intrinsic Geometry of the Cerebral Cortex," *J. of Theor. Biol.* 166, 261-273.

[31] T. Greitz, C. Bohm, S. Holte and L. Eriksson (1991), "A Computerized Brain Atlas: Construction, Anatomical Content and Applications," *J. Comp. Assist. Tomogr.* 15(1):26-38.

[32] B. Hamann (1993), "Curvature Approximation for Triangulated Surfaces," *Computing*, Pages 139-153. Springer-Verlag.

[33] S. C. Joshi, J. Wang, M. I. Miller, D. C. Van Essen and U. Grenander (1995), "On the Differential Geometry of the Cortical Surface," *Proc. of the SPIE Int. Sym. on Optical Science, Engineering, and Instrumentation* San Diego, CA, pp. 304-311.

[34] T. Kapur, E. Grimson, W. M. Wells and R. Kikinis (1996), "Segmentation of Brain Tissue from Magnetic Resonance Images," *Medical Image Analysis*, vol. 1, pp. 109-127.

[35] E. Keppel (1975), "Approximating Complex Surfaces by Triangulation of Contour Lines," *IBM. J. Res. Dev.* 19, 2-11

[36] E. Kreyszig (1959), *Differential Geometry*, Totonto: University of Toronto Press, (Mathematical expositions; no. 11).

[37] W. E. Lorensen and H. E. Cline (1987), "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *ACM SIGGRAPH* Vol 21 No. 4.

[38] D. Luenberger (1989), "*Linear and Nonlinear Programming*," 2nd Ed., Addison-Wesley.

[39] S. D. Ma and A. Gagalowicz (1985), "Determination of Local Coordinate Systems for Texture Synthesis on 3-D Surfaces," *Eurographics'85*, C. E. Vandoni ed. Elsevier Science Publishers.

[40] D. Meyers and S. Skinner (1992), "Surface from Contours," *ACM Trans. Graph.* 11, 3, 228-258.

[41] L. Piegl and W. Tiller (1997), *The NURBS Book,* 2nd Ed., New York, Springer-Verlag.

[42] J. L. Ringo (1991), "Neuronal Interconnection as A Function of Brain Size," *Brain Behav. Evol.* 38, pp. 1-6.

[43] P. E. Roland and K. Zilles (1994), *Trends Neurosci.* 17, pp. 458-467.

[44] J. O'Rourke and V. Subramanian (1991), *On Reconstructing Polyhedra from Parallel Slices,* Tech. Rep. TR 008, Dept. of Computer Science, Smith College, Northampton, Mass..

[45] S. Sandor, R. Leahy and B. Timsari (1996), "Generating Cortical Constraints for MEG Inverse Procedures Using MR Volume Data", In Aine, C.J., F lynn, E.R., Okada, Y., Stroink, G., Swithenby, S.J., and Wood, C.C. (Eds.) *Biomag96: Advances in Biomagnetism Research*, Springer-Verlag, New York.

[46] S. Sandor and R. Leahy (1997), "Surface Based Labelling of Cortical Anatomy Using a Deformable Atlas," I*EEE Trans. on Medical Imaging*, Vol 16, pp 41-54.

[47] E. Schwartz and B. Merker (1986), "Computer-Aided Neuroanatomy: Differential Geometry of Cortical Surfaces and an Optimal Flattening Algorithm," I*EEE Computer Graphics and Applications*, 36-44.

[48] E. L. Schwartz, A. Shaw, and E. Wolfson (1989), "A Numerical Solution to the Generalized Mapmaker's Problem: Flattening Nonconvex Polyhedral Surfaces," *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 11, No. 9. pp. 1005-1008.

[49] P. Shirley and A. Tuckman (1990), "A Polygonal Approximation to Direct Scalar Volume Rendering," *Comput. Graph.*, 24(5):63-70.

[50] K. R. Jr. Sloan and J. Painter (1987), "From Contours to Surfaces: Testbed and Initial Results," *Proc. of CHI + GI' 87* (Toronto, Canada), pp. 115-120

[51] K. R. Jr. Sloan and J. Painter (1988), "Pessimal Guesses May Be Optimal: A Counterintuitive Search Result," *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 6, 949-955.

[52] B. I. Soroka (1981), "Generalized Cones from Serial Sections," *Comput. Graph. Image Proc.* 15, 154-166.

[53] G. Strang (1980), *Linear Algebra and its Applications*, Orlando, Florida 32887: Academic Press Inc., 2nd ed..

[54] J. P. Synder (1987), *Map Projections: A Working Manual, US Geological Survey Professional Paper 1395.*

[55] J. P. Synder (1993), *Flattening the Earth: Two Thousand Years of Map Projections*, University of Chicago Press.

[56] M. R. Spiegel (1959), *Theory and Problems of Vector Analysis and An Introduction to Tensor Analysis*, Schaum Publishing Co. New York.

[57] P. H. Todd (1982), "A Geometric Model for the Cortical Folding Pattern of Simple Folded Brains," *J. Theor. Biol.* 97, pp. 529-538.

[58] P. H. Todd (1986), "The Intrinsic Geometry of Biological Surface Growth," *Lecture Notes in Biomathematics*, Vol. 69 (Levin, S. ed.) Berlin Springer-Verlag.

[59] A. W. Toga, ed. (1990) *Three Dimensional Neuroimaging*, Raven.

[60] D. C. VanEssen and J. H. R. Maunsell (1980), "Two Dimensional Maps of the Cerebral Cortex," *J. Comp. Neurol.*, 191, pp. 255-281.

[61] Y. F. Wang and J. K. Aggrawal (1985), "Construction of Surface Representation from 3D Volumetric Scene Description," *Proc. of Computer Vision and Pattern Recognition*, San Francisco, California.

[62] W. Wells, R. Kikinis, E. Grimson, F. Jolesz (1996), "Adaptive Segmentation of MRI Data," *IEEE Trans. Med. Imaging*, vol. 12, pp. 59-69.

[63] E. Wolfson and E. L. Schwartz (1989), "Computing Minimal Distances on Arbitrary Two-dimensional Polyhedral Surfaces," *IEEE Pattern Anal. Machine Intell.* Vol. 11, No. 9,pp.1001-1005.

[64] A. Worth, N. Makris, J. Meyer, V. Caviness and D. Kennedy (1997), "Automated Segmentation of Brain Exterior in MR Images Driven by Empirical Procedures and Anatomical Knowledge," *Proceedings of the 15th International Conference on Information Processing in Medical Imaging*, pp. 99-112. Springer-Verlag.

[65] M. J. M. Zyda and R. J. Allan (1987), "Surface Construction from Planar Contours," *Comput. Graph.* 11, 4, 393-408.