Short Paper

NEW SURFACE FLATTENING SCHEME AND ITS APPLICATION IN THE VISUALIZATION OF THE HUMAN CORTEX UNFOLDING

Tong-Yee Lee* and Han-Ying Lin

ABSTRACT

It is a difficult problem to interpret the spatial relationship between different locations on the human cortical surface. The main reason for this difficulty is that the surface of the human cerebral cortex is highly folded and therefore much of it is obscured from view. In this paper, we present an efficient and distortion-less flattening algorithm to unfold the cortical surface. Using the proposed technique, we can quickly visualize and precisely measure the cortical surface. Experimental results are shown to evaluate the usefulness of the proposed method.

Key Words: surface flattening, relaxation, cortical surface, visualization.

I. INTRODUCTION

The surface of the human cerebral cortex is a highly folded sheet with the majority of its surface area buried within folds. Much of the human cortical surface is obscured from view by these folds. Therefore, it is difficult to understand the spatial relationship among different locations on the human cortical surface.

The visualization of functional magnetic resonance imaging (fMRI) data is very important to understand the neural activities within the three-dimensional folds of the human brain. Surface flattening is an important and widely used technique to visualize and measure the brain surface. To precisely analyze the brain function, the flattened representations of 3D brain surface must preserve several vital spatial relationships such as angle, area and distance, among regions on the cortical surface. In the past, there have been many techniques presented to flatten or unfold the three-dimensional brain cortex (Schwartz and Merker, 1986; Carman *et al*., 1995; Drury *et al*., 1996; Thompson and Toga, 1996; Wandell *et al*., 1997; Fischl *et al*., 1999; Wandell *et al*., 2001) onto two-dimensional space. However, most techniques are very computationally expensive. Their computational speeds vary from minutes to hours. In contrast to these previous works, our approach usually takes only a few seconds (Fischl *et al*., 1999; Wandell *et al*., 2001).

In this paper, we present a novel technique to flatten 3D surfaces. We apply this new technique to the visualization and measurement of the human cerebral cortex. The proposed method is very simple to implement and it can flatten a cortical surface in a few seconds with little distortion. The standard deviation of an area ratio is below 0.1 in our experiment and this result is comparable to other work (Fischl *et al*., 1999; Wandell *et al*., 2001).

Our paper is organized as follows. The relaxation-based flattening method is presented in Section II. In Section III, the proposed flattening method is employed to visualize and measure a cortical surface. Finally, we conclude our paper in Section IV.

II. FLATTENED MAP CONSTRUCTION

1. Overview

Our proposed flattening technique can handle a

*Corresponding author. (Tel: 886-6-2757575 ext. 62531; Fax: 886-6-2747076; Email: tonylee@mail.ncku.edu.tw)

The authors are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan 701, R.O.C.

two-manifold and genus-0 3D mesh that is topologically equivalent to a disk. If the given 3D meshes are not genus-0, an extra cutting is necessary to partition meshes into several genus-0 3D patches. The original 3D object is a polyhedral mesh with triangular faces. To create a flattened map of a 3D object, the following steps are executed.

First, the users select n control vertices on a given 3D object. We use Dijkstra's shortest path algorithm to find each path connecting consecutive control vertices. These paths are used to partition the input meshes into several 3D patches. Second, we initially flatten each 3D patch onto an n-side regular 2D flattened map. Finally, we use an efficient relaxation method to further adjust the vertices on the flattened map. More details about the second and third steps will be described in the following subsection.

2. Flattening Algorithm

For a given 3D patch and n control vertices along this 3D patch boundary, we flatten this 3D patch onto an n-side regular 2D polygon called F (i.e., flattened map). This n-regular polygon F is inscribed in the unit circle and its center is at (0,0). These n control vertices are mapped to the vertices of F and each path connecting consecutive control vertices is mapped to an edge of F. The 2D coordinates of those non-control vertices along each path are interpolated based on the arc length of the path. Then, we compute the 2D coordinates of the interior vertices. Initially, these non-boundary vertices are all mapped to the center position (0, 0). Then, these interior vertices are moved step by step by the following relaxation equation.

$$p_i' = (1-\lambda)p_i + \lambda \frac{\sum_{j=1}^{k_i} (\omega_j p_j)}{\sum_{j=1}^{k_i} \omega_j} \tag{1}$$

where

- k_i : the number of vertices connected to a vertex i in the original model.
- λ : a parameter controls the movement speed and its value is between 0 and 1.
- p_j : the mapped 2D coordinates of a vertex j connected to a vertex i.
- w_j : pulling weight for p_j (see Fig. 1).
- p_i' : the newly mapped 2D position of p_i after an iteration.

Each interior vertex p_i is moved by its 1-ring neighboring vertices p_js step by step. This process

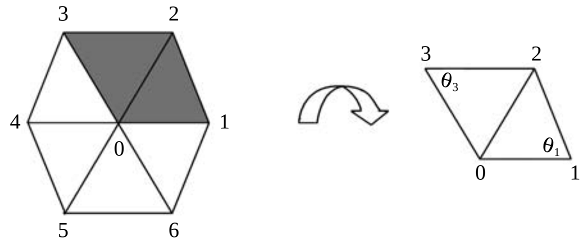

Fig. 1 The determination of the pulling weights of 1-ring neighbors

will continue until all of the interior vertices are stable. On the other hand, the boundary vertices are all fixed in the whole process. To preserve the aspect ratio of the original triangle versus the mapped triangle (i.e., not cause too much distortion), we use Eq. (2) to determine w_j. Fig. 1 shows an example of pulling weight w_2 of a p_2 and p_i is p_0. In Eq. (2), if θ_i, i = 1, 3 tends to very small (i.e. a very narrow triangular mesh), the value of w_2 can be an infinite value. However, in general, this extreme case does not occur in medical applications. In medical applications, the triangular meshes are usually generated from a standard marching cube algorithm. The marching cube algorithm always generates a triangular mesh with a regular structure. Therefore, the proposed method can work very well in medical applications.

$$w_2 = \cot\theta_1 + \cot\theta_3 \tag{2}$$

Based on Eq. (1), the iteration method can be used to find all 2D coordinates of the interior vertices. However, in this manner, the computation time is not predictable and could be expensive. There is an obvious way to speed up computing Eq. (1). For each interior vertex p_i, we measure its displacement between consecutive iterations. Then, we average all $p_i s$' displacements and we set a threshold on the difference of the average displacement between consecutive iterations. If this difference is below a selected threshold, we terminate the iteration process. However, in this manner, the number of iterations is still unpredictable using this naïve approach.

Using Eq. (1), as p_i is stable, ideally, $p_i' = p_i$. Thus, we will have the following form.

$$p_i' = p_i = (1-\lambda)p_i + \lambda \frac{\sum_{j=1}^{k_i} (\omega_j p_j)}{\sum_{j=1}^{k_i} \omega_j}$$

$$\Rightarrow p_i = \frac{\sum_{j=1}^{k_i} (\omega_j p_j)}{\sum_{j=1}^{k_i} \omega_j} \tag{3}$$

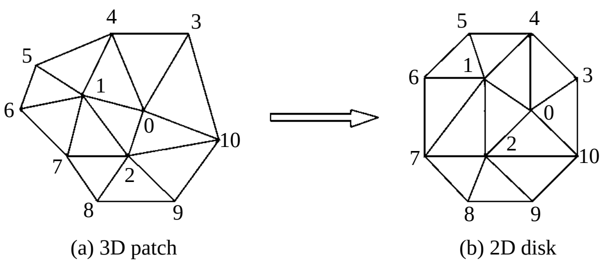

Fig. 2 An example of flattening a 3D patch onto an 8-sided regular 2D disk

Let $X_i = \sum_{j=1}^{k_i} \omega_j$, $i = 1 \sim N$. In the following, we use Fig. 2 as an example to explain Eq. (3) and to demonstrate how a linear system can be used for the proposed relaxation method. In Fig. 2, the Eq. (3) can be represented by (4).

$$\begin{cases} \chi_0 p_0 = \omega_{01}p_1 + \omega_{02}p_2 + \omega_{010}p_{10} + \omega_{03}p_3 + \omega_{04}p_4 \\ \chi_1 p_1 = \omega_{10}p_0 + \omega_{14}p_4 + \omega_{15}p_5 + \omega_{16}p_6 + \omega_{17}p_7 \\ \qquad + \omega_{12}p_2 \\ \chi_2 p_2 = \omega_{20}p_0 + \omega_{21}p_1 + \omega_{27}p_7 + \omega_{28}p_8 + \omega_{29}p_9 \\ \qquad + \omega_{210}p_{10} \end{cases} \tag{4}$$

To solve this linear system, we can split it into x and y components in the following forms:

$$\begin{bmatrix} \chi_0 & -\omega_{01} & -\omega_{02} \\ \omega_{10} & -\chi_1 & \omega_{12} \\ \omega_{20} & \omega_{21} & -\chi_2 \end{bmatrix} \begin{bmatrix} p_{0x} \\ p_{1x} \\ p_{2x} \end{bmatrix} = \begin{bmatrix} \omega_{010}p_{10x} + \omega_{03}p_{3x} + \omega_{04}p_{4x} \\ -(\omega_{14}p_{4x} + \omega_{15}p_{5x} + \omega_{16}p_{6x} + \omega_{17}p_{7x}) \\ -(\omega_{27}p_{7x} + \omega_{28}p_{8x} + \omega_{29}p_{9x} + \omega_{210}p_{10x}) \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \chi_0 & -\omega_{01} & -\omega_{02} \\ \omega_{10} & -\chi_1 & \omega_{12} \\ \omega_{20} & \omega_{21} & -\chi_2 \end{bmatrix} \begin{bmatrix} p_{0y} \\ p_{1y} \\ p_{2y} \end{bmatrix} = \begin{bmatrix} \omega_{010}p_{10y} + \omega_{03}p_{3y} + \omega_{04}p_{4y} \\ -(\omega_{14}p_{4y} + \omega_{15}p_{5y} + \omega_{16}p_{6y} + \omega_{17}p_{7y}) \\ -(\omega_{27}p_{7y} + \omega_{28}p_{8y} + \omega_{29}p_{9y} + \omega_{210}p_{10y}) \end{bmatrix} \tag{6}$$

$$\Rightarrow \begin{cases} \boldsymbol{AX} = b_x \\ \boldsymbol{AY} = b_y \end{cases} \Rightarrow (\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{A}^{-1}b_x, \boldsymbol{A}^{-1}b_y) \tag{7}$$

Usually, the vantage (i.e., degrees) of a vertex on a 3D model is a small integer varying from 2 to 7. As the number of interior vertices becomes larger and larger, the matrix $\boldsymbol{A}$ in Eq. (7) will become a sparse matrix. Floater (1997) proves that when $\chi_i = 1$ and the boundary vertices are mapped onto a convex polygon, there is a unique solution for Eq. (7). Our flattening algorithm satisfies these two constraints. In the proposed method, the n-sided regular flattened map F is a convex polygon. To solve a sparse linear system is more efficiently than to compute results by the simple iteration method. In the implementation, we employ a biconjugate gradient method (William *et al*., 1992) with a computational complexity approximate $O(N)$ (i.e., N is the number of p_i) to solve this large, sparse linear system.

2. Measurement

In this section, we describe how to evaluate the quality of the flattened map. Like most previous works, for example in (Fischl *et al*., 1999; Wandell *et al*., 2001), the distortion-less flattening means that the aspect ratio the original 3D surface versus the flattened 2D surface must be preserved. Following other works (Fischl *et al*., 1999; Wandell *et al*., 2001), we use the standard deviation of an area ratio and the histogram of an area ratio to measure the quality of flattening. To compute the histogram, we have the following arrangements. Assume the area ratio varies from $R_{\min}$ to $R_{\max}$. For the histogram, we map $[R_{\min}, R_{\max}]$ onto the interval $[X_{\min}, X_{\max}]$ of the X-axis. For the Y-axis, we use the number percentage of area ratios that are located at the same subinterval. In Section III, we will use this measurement to evaluate the human cortical surface after flattening the cortex.

III. CORTICAL SURFACE FLATTENING AND VISUALIZATION

It is a difficult task to analyze the location and size of functional brain activity across subjects. The major reason is that individual differences in folding patterns and functional foci are often obscured from view. Flattening is a tool to convert the 3D cortical surface into a simple 2-D sheet topology that is easy to measure. After a 3D-to-2D flattening process, many interesting spatial relationships in functional and anatomical data can be discovered. Our test cortical surface is represented as a triangular mesh that was created from a standard marching cube algorithm. In this Session, we apply our relaxation method to flatten this cortical surface. Fig. 3(a) shows a 3D view of this cortical surface. We obtained this dataset from Dr. Paul Bourke at Astrophysics and Supercomputing Center, School of Biophysical Sciences and Electrical Engineering, Swinburne University of Technology in Melbourne, Australia. In this example, we partition this dataset into three portions: (b) (c) (d) for easy examinations. Note that in this

Table 1 Experimental statistics of flattening three parts of the cortical surface

Info / Patch	Vertex number	Edge number	Triangle number	Flattening time	Area ratio (standard deviation)
Upper part	777	2211	1431	0.37 sec.	0.06
Middle part	1578	4589	2998	0.62 sec.	0.09
Lower part	1504	4368	2857	0.54 sec.	0.05

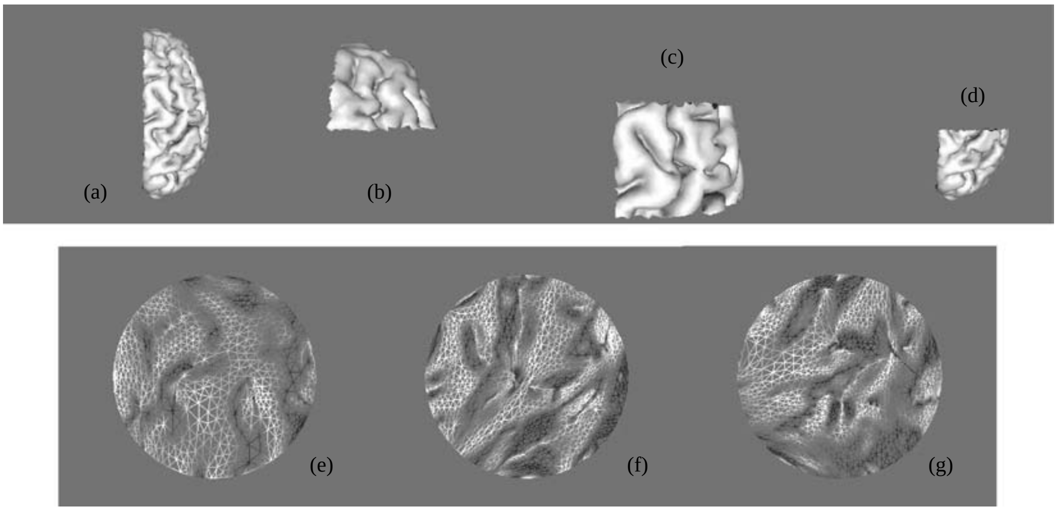

Fig. 3 Visualization of the cortical surface

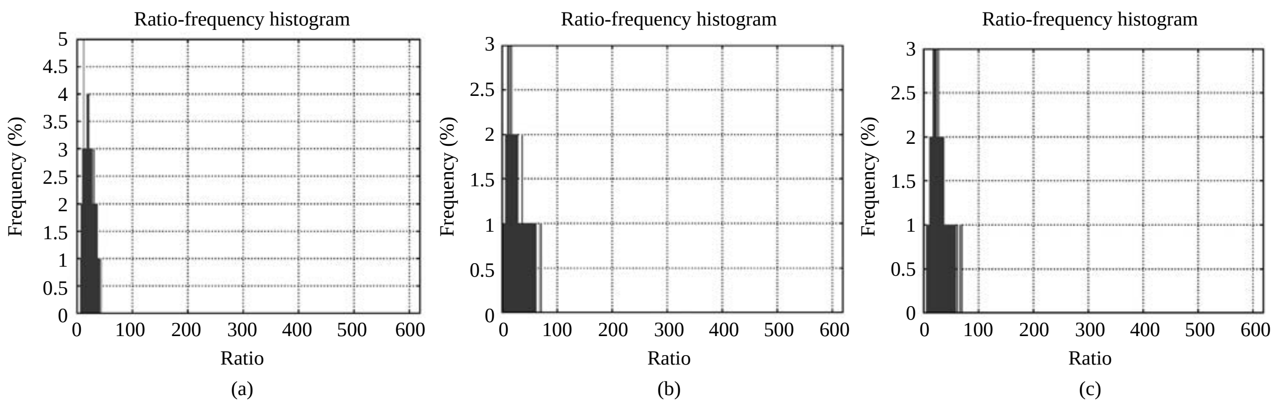

Fig. 4 Histograms for three parts of the cortex in Fig. 3; (a) the upper part, (b) the middle part, and (c) the lower part

application the examiners are particularly interested in visualizing the folded regions. We attempt to use the curvature information to visualize these folded regions. These folded regions always have high curvatures in contrast to other flat regions. They are always obscured from view. Fig. 3(e), (f) and (g) show the flattened maps for (b), (c) and (d). In these three flattened maps, it is easy to identify these folded regions, since these folded regions are encoded in a darker color by the curvature information.

We use the metric described in Section II.3 to evaluate the flattened maps of the example in Fig. 3. In Fig. 4, each histogram corresponds to the measurement for each portion of cortex in Fig. 3. Table 1 shows statistics of geometrical information, experimental timings and the standard deviation for the area ratio in this experiment. Our algorithm was performed on a PC with a Pentium III 1G and 256MB. For the cortex visualization, the examiners, in particular, want to measure the area of cortical surface. From both Fig. 4 and Table 1, we can observe that the quality of our flattening is very good.

The standard deviation for an area ratio is very small and it is below 0.1. This result is comparable to previous work (Fischl *et al*., 1999; Wandell *et al*., 2001). However, our computational speed is very fast compared to (Fischl *et al*., 1999; Wandell *et al*., 2001). Fischl *et al*.'s (1999) flattening procedure typically requires on the order of 15 hours to flatten a full cortical hemisphere on a 266 MHz Pentium. In Wandell *et al*.'s work (2001), the processing time on a 300 MHz Pentium can be as long as two days. Algorithm speed can be an important problem in (Fischl *et al*., 1999; Wandell *et al*., 2001) if interactive exploration and visualization is required. For the page limit, more experiments and examples can be found in (Lin, 2001) and http://couger.csie.ncku.edu.tw/~vr/cortex_flattening.html.

IV. CONCLUSION AND FUTURE WORK

Surface flattening is a very powerful technique. It is widely applied to many areas. In this paper, we propose a simple but an effective and distortion-less relaxation method to flatten 3D cortical surface. Furthermore, we propose to solve the relaxation method by a linear, sparse system instead of a naïve iteration method. From the experimental results, the proposed method does not cause much distortion on the flattened maps. The standard deviation for an area ratio is below 0.1 in our experiments. This result is good and is comparable to previous work. Remarkably, the proposed method can flatten the cortical surface in a few seconds. This speed can significantly help medical examiners to correctly and quickly measure the area of the cortical surface. There are many future possibilities with this approach. For example, the proposed method can work very well for a triangular mesh with a regular structure. For extreme cases, such as very narrow triangles, the weighting function can have an infinite value. Therefore, this extreme case might produce an intersection of polygons. Although these extreme cases are very rare in medical applications, we will design another alternative for this weighting function to deal with the extreme case. In addition, we are also surveying the possibility of applying the proposed method to other important areas such as texture mapping in computer graphics. These extensions will be done in the near future.

ACKNOWLEDGMENTS

This project was supported by the National Science Council, Taiwan, Republic of China, under contract No. NSC-93-2213-E-006-026 and NSC-93-2213-E-006-060.

REFERENCES

Carman, G. J., Drury, H. A., and Van Essen, D. C., 1995, "Computational Methods for Reconstructing and Unfolding the Cerebral Cortex," *Cerebral Cortex*, Vol. 5, No. 6, pp. 506-517.

Drury, D. van, Anderson, E. C., Lee, C., Coogan, T., and Lewis, J., 1996, "Computerized Mappings of the Cerebral Cortex: A Multiresolution Flattening Method and a Surface-based Coordinate System," *The Journal of Cognitive Neuroscience*, Vol. 8, No. 1, pp. 1-28.

Fischl, B., Sereno, M., I., and Dale, A. M., 1999, "Cortical Surface-Based Analysis," *NeuroImage*, Vol. 9, No. 2, pp. 195-207.

Floater, M. S., 1997, "Parametrization and Smooth Approximation of Surface Triangulations," *Computer Aided Geometric Design*, Vol. 14, No. 3, pp. 231-250.

Lin, H. Y., 2001, "Three-Dimensional Polyhedron Flattening Techniques and Its Applications," *Master Thesis*, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1992, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK.

Schwartz, E. L., and Merker, B., 1986, "Computer-Aided Neuroanatomy: Differential Geometry of Cortical Surfaces and an Optimal Flattening Algorithm," *IEEE Computer Graphics and Applications*, Vol. 6, No. 3, pp. 36-44.

Thompson, P. M., and Toga, A. W., 1996, "A Surface-Based Technique for Warping 3-Dimensional Images of the Brain," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 4, pp. 402-417.

Wandell, B., Chial, S., and Backus, B. T., 2001, "Visualization and Measurement of the Cortical Surface," *The Journal of Cognitive Neuroscience*, Vol. 12, No. 5, pp. 739-752.

Wandell, B., Engel, S., and Hel-Or, H., 1997, "Creating Images of the Flattened Cortical Sheet," *Investigative Ophthalmology & Visual Science*, Vol. 37, No. 3, pp. 1081.

Manuscript Received: May 20, 2002
Revision Received: Mar. 10, 2003
and Accepted: Apr. 14, 2004