<u>Modified Normal Vector Voting Estimation in neuroimage</u>

Neuroimage Processing (339.632)
Instructor: Moo. K. Chung

Seung-goo KIM
Dec 15, 2009

Abstract

The normal vector is one of the metrics that are sensitive to the property of the curved surface. In neuroimage studies, normal vectors are used to strip skulls, to analyze cortical thickness, to assess the principle direction of diffusion tenor and to estimate the location of neural activities. However, conventional methods to estimate the vertex normal on the tessellated mesh are sensitive to irregular triangulation and noise signals occurred from the previous processing to construct a cortical surface model form MRI volume data. In the present project, an algorithm is proposed to estimate a normal vector on a vertex in a more precise fashion. Following the idea Normal Vector Voting algorithm [1], Modified Normal Vector Voting algorithm is implemented. Simulations on a sphere and a complex surface were performed with various amount of noise to assess the proposed method comparing to conventional ones. The errors of estimation from the proposed method were the least in all performed cases demonstrating the robust estimation to noise of the proposed algorithm. The large amount of computational load of the proposed algorithm remains as a further problem for practical application.

1. Introduction

1.1. Normal vector in neuroimage

The normal vector of surface is defined as a vector perpendicular to it [2]. The normal vector is one of the metrics that are sensitive to the property of the curved surface. The most common usage of normal vector in computer-aid graphic processing is about the lighting effect. It is also true for the neuroimage domain since the lighting effect algorithm based on normal vectors is routinely applied in 3-D rendering tool.

For the purpose of scientific analyses, the traditional usage of normal vector in neuroimage studies is to define the cortical thickness as the length from the vertex on the inner (outer) cortical surface toward the inner (outer) cortical surface along the surface normal on the corresponding vertex [3]. Another application of normal vectors is in the popular skull striping application Brain Extraction Tool (BET) [4]. It uses surface normal vector as a direction of step-wise iterative deformation. Each vertex push toward the inward surface normal so that the surface would be smooth and all vertices equally would be spaced.

The recent application of normal vectors is in a diffusion tensor imaging (DTI) study, to obtain the information about how the diffusion tensor is aligned to the surface in developing human fetal brains, the difference between principle eigen vector and the surface normal on the triangular brain surface model was computed in terms of angel [5].

Besides structural analyses, the normal vector on a vertex in a brain model mesh is also used for functional data analysis [6]. To estimate the distributed current sources of measured magnetic field using magnetoencephalographic (MEG) instruments, it is rational to constraint the possible space of the current source as the intermediate surface between outer and inner cortical surfaces because the most affective neural electromagnetic events to MEG are known as excitatory postsynaptic potential (EPSP) or postsynaptic potential (IPSP) of the pyramidal cells in the laminar IV of the cerebral cortex. Dendrites of pyramidal cells are mostly aligned to the direction of the normal vector on the cortical surface. For this reason, in the solution called Minimum-Norm Estimation (MNE), the possible current sources are presumed as to be within the cortical ribbons directing the normal vector, and the neural activities over the whole brain are modeled as several normal vectors on the surface mesh, spacing regarding the spatial resolution of MEG instrument. In this sort of model, it is critical issue to construct an accurate surface model as it enhances the results for the location of neural activities.

1.2. Related methods and motivation

The surface normal can be obtained by the cross product of two non-collinear vectors both in a plane. Also for a curved surface, the normal vector on a certain point within the surface can be defined as a vector perpendicular to the tangential plane to the point. If we know the equation of the curved surface, then we can get the equation of the tangential plane deriving the surface, and analytically can obtain the normal vector on a point.

The reality is that only things we handle in computerized neuroimage is a piecewise-smooth tessellated triangular mesh as an approximation of the curved surface in the real world. This challenge to estimate normals on a curved surface via a polygonal approximation is well known in the computer graphic field and also vastly investigated for a long time. Henri Gouraud had implemented an algorithm for shading effect on polygons using estimation of vertex normals in 1971 [7][1]. Since then various approaches have seen

[1] This approach is quite simple in where the estimation for a vertex normal is an arithmetic average of surface normals of surrounding facets. But for this simplicity, the major advantage of this approach

introduced [8].

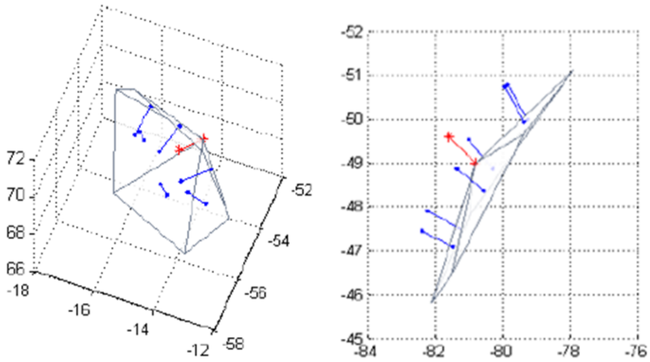

Fig. 1. Examples of normal estimations on a vertex (red) and surface normals (blue) of neighboring triangles

One idea is to set weighting parameters in average in aspect of the property of neighboring triangles rather than simply setting all as one as Gouraud. Taubin proposed the areas of triangles [9], and Thurmer and Wuthrich proposed the interior angle of triangles [10] as the weighting factors. Let $\widehat{\boldsymbol{N}_i}$ is estimation for the normal on the *i-th* vertex, then the each estimation can be formalized as

$$\widehat{\boldsymbol{N}_i} = \frac{\sum_{j=1}^{m} \boldsymbol{\varphi}_j \boldsymbol{n}_j}{\sum_{j=1}^{m} \boldsymbol{\varphi}_j} \quad \text{for Taubin's method}$$

and

$$\widehat{\boldsymbol{N}_i} = \frac{\sum_{j=1}^{m} \boldsymbol{T}_j \boldsymbol{n}_j}{\sum_{j=1}^{m} \boldsymbol{T}_j} \quad \text{for Thurmer and Wuthrich's method}$$

where $\mathbf{m}$ is the number of neighbor triangles, $\boldsymbol{n}_j$ is the surface normal of the *j-th* triangle, $\boldsymbol{\varphi}_j$ is the angle of the *j-th* triangle, and $\boldsymbol{T}_j$ is the area of the *j-th* triangle.

Alternately, Principle Component Analysis (PCA) can be used to estimate the normal vector [11]. To briefly address the idea, the eigen vector of the covariance matrix of the coordinates of the center vertex and the sets of vertices embedded neighboring triangles would be orthogonal bases which represent the variance of points the best. Assuming those points forms a curved surface, the eigen vector with the least eigen value would be an approximation of the normal on the center vertex. Formal explanation is stated in [11].

Also there may be other methods such as linear regression and finite difference method as reviewed in [8]. But those are not of interest in this project.

1.3. Problems and aim statement

To estimate the normals using facets normals or via PCA may be biased in cases such as

(1) when the triangles in mesh is irregular

Triangles in the mesh should be different in the length of edges and thus areas to fit the curved surface while each triangles are still plane, i.e. piecewise-smooth. So there cannot be ‘perfect regularity’ in triangles. However, irregular tessellation bias the normal vector estimation especially based on the facet normals. In the examples from [1], the mesh in (b) shows irregular triangulation leads to the biased coverage of neighbor triangles. In other

---

is the fast computation which can be critical dealing with massive numbers of polygons in a domain such as 3D graphic games.

words, when the triangulation is irregular, the discrepancy between the area of 1-ring[2] neighbors (**Fig 2**; alternately shaded triangles right surrounding the open circle in the center) and the area within a certain geodesic distance (**Fig 2**; black thick circle) would increase.

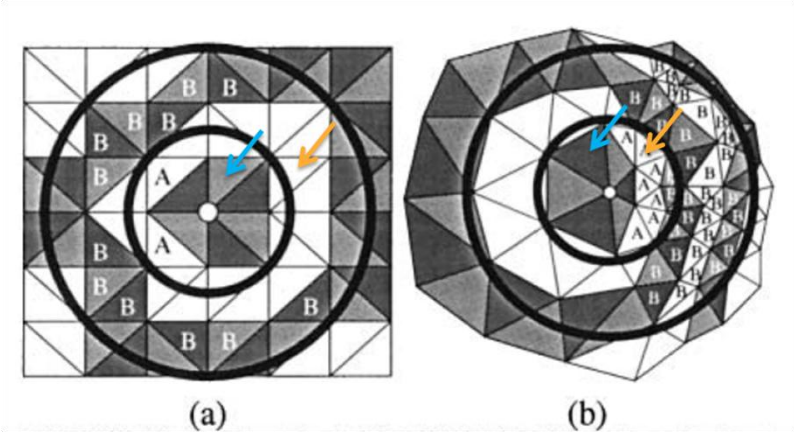

Fig. 2. (a) an example of regular triangulation and (b) irregular triangulation; Reprinted from [1]

Irregular triangulation is also in case with the example of cortical surface model from [12]. The cortical model is constructed by a deformable surface algorithm. In this algorithm, the mesh with 40,962 vertices and 81,920 triangles starts as an ellipsoid outside of the brain, and it shrinks in a stepwise fashion with until it reach the segmented white-matter volume in MRI image to construct the inner cortical surface, which is more easily detectable than the interface between gray matter or pia matter, which is a thin tissue that covers the outer surface of cerebral cortex, and cerebrospinal fluid (CSF). Then the surface expands with constraints to construct the outer surface, or the pial surface.

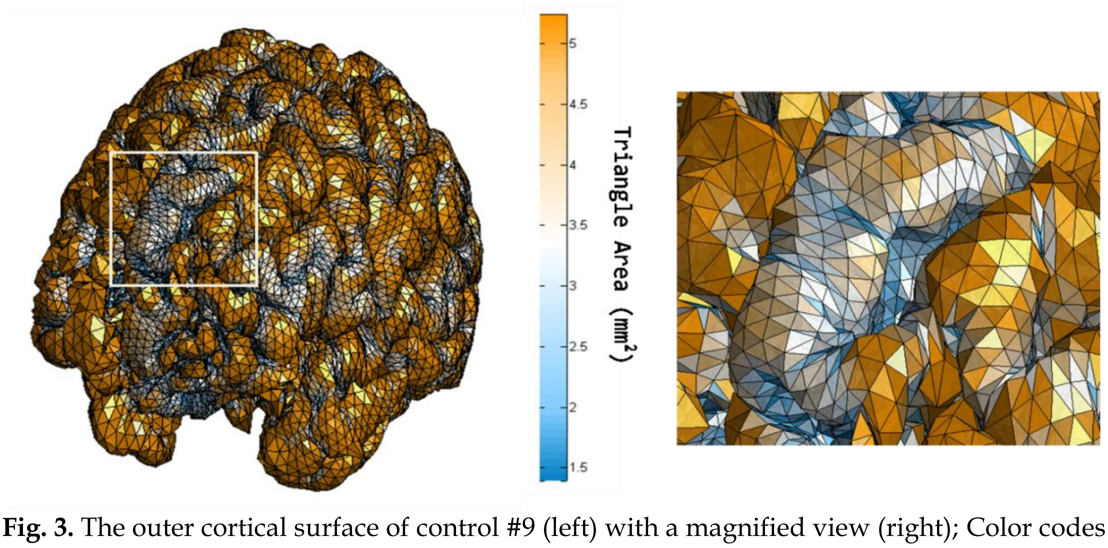

Fig. 3. The outer cortical surface of control #9 (left) with a magnified view (right); Color codes represent the area of triangles.

The outer cortical surface of the normal control subject #9 is visualized (**Fig. 3**). As it can be seen in the magnified view, the triangles in the mesh of cortical surface model used in practical researches shows irregularity, especially on the sulcal area where the surface is more complicated than gyral areas.

---

2 *K-ring* refer to the set of triangles in the *K-th* ring around the center vertex. The alternately shaded triangles(Fig.2; indicated by a blue arrow) are 1-ring neighbors. And white triangles surrounding 1-ring(Fig.2; indicated by a orange arrow) are 2-ring neighbors. See Page et al (2002) for details.

(2) when the mesh contains noisy points

The second case when the normal estimation can be biased is about the noise. Because in the deformable surface algorithm, a vertex on the original ellipsoid may not be 'pushed' enough due to the errors from the previous segmentation on the MRI volume data. That is, if a voxel in the CSF is classified as a white matter, then the corresponding vertex left outside the cortical surface; this leaves a 'throne' on the surface [13]. In case of the autism subject #1 from [12], the outer cortical surface model has a several 'thrones' especially the ventral region of the temporal lobe and occipital lobe. This noise is possibly due the detachment of the brain stem and cerebrum (**Fig 4**).

The cases described above are well noticeable in the histogram (**Fig. 5**). The aim of this project is to propose an algorithm which can estimate the normal vector on a vertex despite of irregularity in triangulation and noise on the mesh from MR image processing and its implementation to compare the performance with congenital methods.

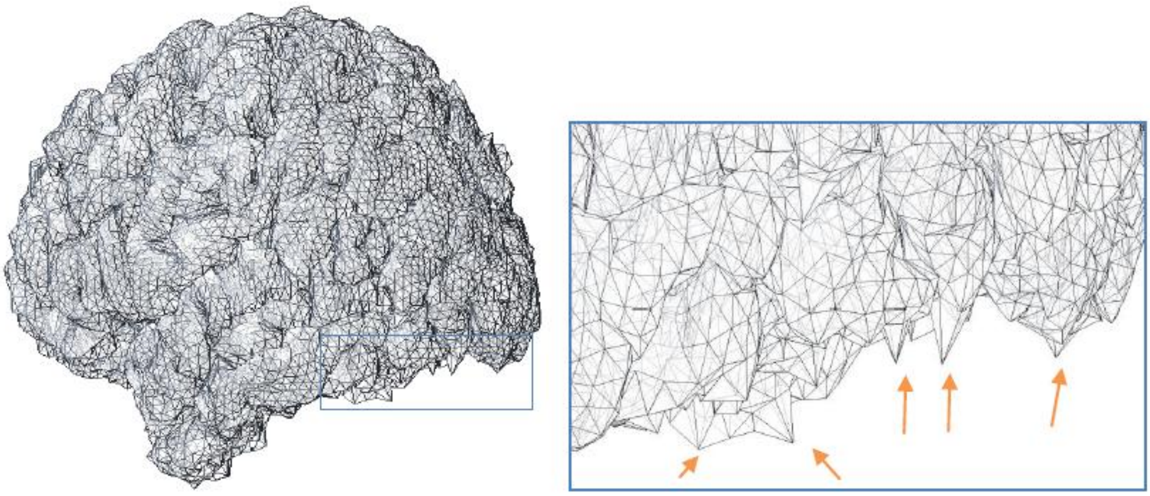

Fig. 4. The outer cortical surface of the autism subject #1 (left); Vertices with noise indicated with orange arrows in a magnified view (right)

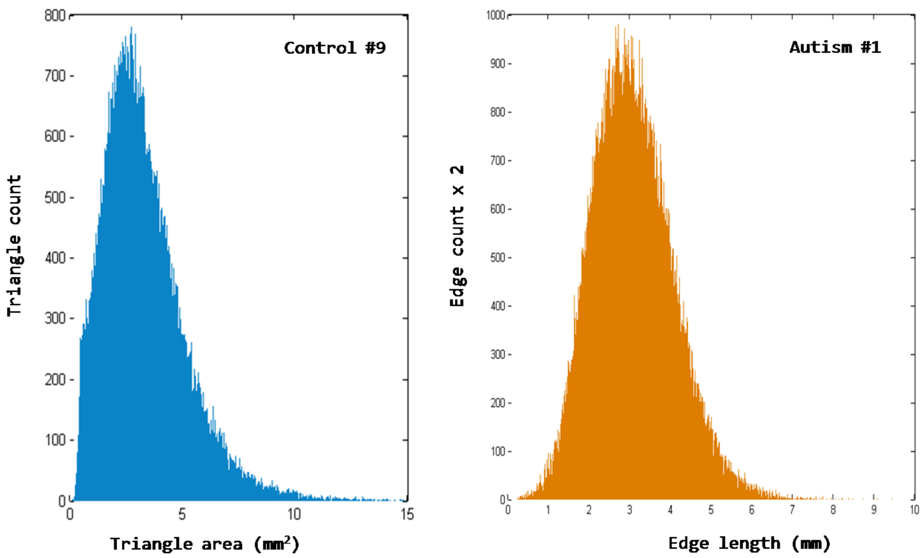

Fig. 5. Histograms of the areas of triangles (left; control #9) and the length of edges (right; autism #1) in outer cortical surface models

2. Proposed method

2.1. Normal Vector Voting algorithm

The idea of Normal Vector Voting to estimate the vertex normal on triangular mesh was introduced by Page and colleagues [1]. To briefly explain the procedure; (1) find the neighbors defined in terms of geodesic distance, (2) get 'votes' from each neighboring triangles, (3) perform eigen analysis to extract the agreement amongst votes, (4) finally estimate the orientation of the normal and curvature. Refer to [1] for details.

The key innovation of this algorithm is to define the neighbors in terms of geodesic distance to balance the range of neighbor even on the irregularly triangulated mesh and to penalize the 'vote' from a distant triangle under the assumption of that the curvature on a curved surface changes continuously. In other words, the more the neighbor is nearby, the more the neighbor is likely to know about the center vertex normal. What the 'vote' here refers to is a certain transformation ($\boldsymbol{N_i}$) of the individual surface normal ($\boldsymbol{N}$) of each neighboring triangle ($\boldsymbol{f_i}$) as one of the normal estimation for the vertex ($\boldsymbol{v}$) (**Fig. 6**).

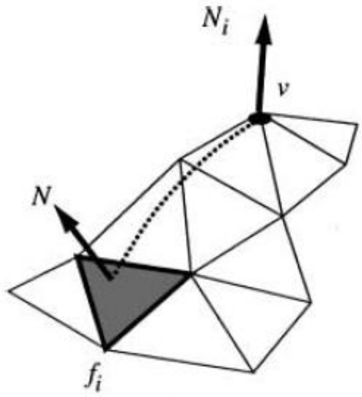

Fig.6. Voting scheme; Reprinted from [1]

Another innovative part is the eigenvalues analysis, rather than averaging the votes. It is possible to detect the crease or corner by classifying the tensor eigenvalues. There is a dominant eigen value in the covariance matrix of votes when the vertex of interest is on the smooth surface, whereas the preference decreases when the vertex is near the crease or corner.

However, in the present context, it is not necessary to detect the crease or junction of surfaces handling the cortical surface models. It would be natural to assume the cortical surface as a continuous curved surface, and an orientble one since it is constructed from an ellipsoid. For the simplicity and reduced computational load, the eigen value analysis is modified as weighted averaging in the present implementation.

Another modification is using Euclidian distance instead of geodesic distance. One reason is that computing geodesic distance in 3D space itself requires a huge amount of computational load, and also highly complicated problem with a manifold with a large number of vertices. On the other hand, computing Euclidian distance in 3D space can be done as the L2norm of the vector between the points costing a little resource. Besides, the boundary defined in this implement is relatively small, and it is not likely to have critical bias except inside extremely narrow fissure. For those reasons, using Euclidian distance is assumed to yield a reasonable computational load and result.

2.2. Modified Normal Vector Voting algorithm

The main ideas of the present algorithm are to take more neighbors than simple 1-ring, and to modify the surface normals to fit the vertex normal better, and to penalize the votes from distant triangles.

The steps of proposed algorithm are

(1) define the 2-ring neighbors surrounding the center vertex
(2) cast votes from neighbors
(3) average the votes weighting by respective area and the Euclidian distance.

The details of each step are explained as following.

2.2.1. 2-ring neighbor definition

The neighbors are defined as 1-ring (**Fig. 7**; triangles colored by cyan) and 2-ring neighbors (**Fig. 7**; triangles colored by magenta). Since the average length of edges are around 3mm in the data set, the expected radius of this 2-ring is 6mm. This small range of neighbors makes using Euclidian distance in this implementation instead of geodesic distance. If the boundary of neighbors is too large, then the discrepancy between Euclidian distance and geodesic distance on a manifold would be significantly large. Also this boundary makes the computational load reasonable. The issue will be discussed later.

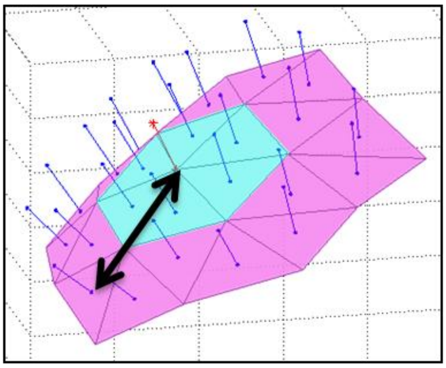

Fig. 7. 2-ring neighbors surrounding a certain vertex (marked by a red asteroid)

2.2.2. Vote casting

Unlikely other weighted averaging methods, in Normal Vector Voting algorithm and the present implementation, the 'opinion' ($\boldsymbol{N_i}$) about the vertex normal of the individual surface ($\boldsymbol{f_i}$) is not the surface normal of itself (**Fig. 8. a**). Instead, the 'vote' is the surface normal rotated by ($\boldsymbol{2\theta - \pi}$) in plane $\boldsymbol{\Pi_i}$ which contains both surface normal ($\boldsymbol{N}$) and its vote ($\boldsymbol{N_i}$) (**Fig. 8. b**). The reason of this rotation is that the underlying surface between the centroid of the triangle and the center vertex would be curved. The casted votes can be more likely to be similar with the real normal on the center vertex in this fashion.

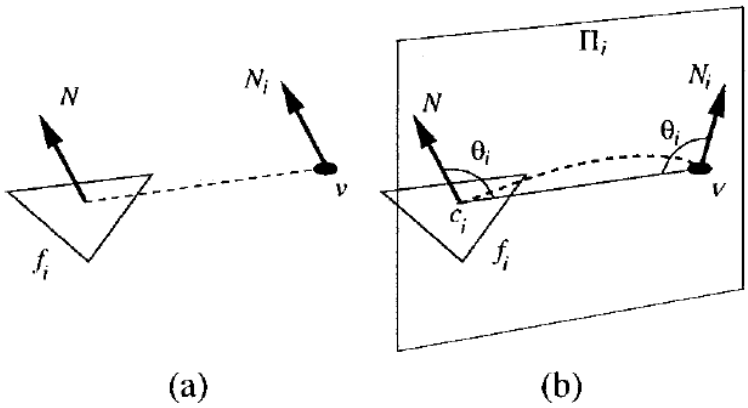

Fig. 8. Voting scheme; Reprinted from [1]

2.2.3. Vote collection

The last step for the estimation of normal on *j-th* vertex ($\widehat{N_v^j}$) is to average the votes as

$$\widehat{N_v^j} = \sum_{i=1}^{m} N_i W_i^j$$

$$W_i^j = \frac{T_i}{\sum_{i=1}^{m} T_i} exp(-2\frac{\|\overrightarrow{vc_i}\|}{\mathbb{E}(e_l)})$$

where N_i is an individual vote from the *i-th* neighboring triangle, m is the number of triangles within 2-ring boundary, W_i^j is the weighting factor, T_i is the area of i-th triangle, $\|\overrightarrow{vc_i}\|$is the Euclidean distance as the L2norm of the vector between the *j-th* vertex and the centroid of *i-th* triangle (**Fig. 7**; two headed black arrow), and $\mathbb{E}(e_l)$ is the expected length of edge as the average of all edges in the whole mesh.

Note that the weighting factor penalizes the corresponding vote exponentially. The reason for this discounting, or decaying, is the assumption of the continuous change of the curvature. While assuming no abrupt change of curvature included in the surface of interest, i.e. the brain, the facets located further from the center vertex would be more likely to have a different normals from the normal that is to be estimated. The reason for that the decaying function is exponential is not clearly explained in the original paper of Page et al [1]. However, it can be reasoned from the fact that, as the boundary of neighbors increases, the number of neighbors increase exponentially. If the decaying function is linear, the sum of votes on the iso-distance contour cannot be stable across the distance. It would be rational to use an exponential function to control the distribution of the sum of each distance consistent.

3. Simulation

3.1. Comparison scheme

To compare the performance of the present algorithm related to other conventional ones, following steps were performed.

(1) Synthesize a smooth surface
(2) compute vertex normals ($\widehat{N^0}$) with a certain method on it
(3) add a certain amount of Gaussian noise on the surface
(4) compute vertex normals ($\widehat{N^1}$) with the same method
(5) compute the estimation error on the i-th vertex defined as

$$error_i = acos\frac{< \widehat{N_i^0}, \widehat{N_i^1} >}{\|\widehat{N_i^0}, \|\|\widehat{N_i^1}\|}$$

Note that the estimation error is defined in terms of the angle between $\widehat{N^0}$ and $\widehat{N^1}$. Since the manifold used in this simulation is not defined from surface equation but synthesized from the real data, the ground truth for the vertex normal cannot be derived analytically. However, it is still possible to compare the change of estimations occurred by noise addition, and it is defined as estimation errors in this sense. It is for the surface of interest is the human brain in this project. However, to validate the performance with the ground truth, the surface model of a unit sphere is also used.

To compare the performance of the present algorithm related to other conventional, Taubin's method (averaging facet normals weighted by area) and PCA were performed along with the proposed method, the Modified Normal Vector Voting algorithm (simply referred as ‘Voting’ henceforth).

3.2. Objects: unit sphere and cortex-like surface.

The objects used in this simulation are a unit sphere and a cortex-like surface (**Fig. 9**). Note that the used sphere model is not an ideal sphere but a 81,920-facet polygon as a discrete approximation. The pentagonal pattern of areas of the sphere model is a byproduct of the constructing method of subdividing the surface. The cortex-like surface is constructed by smoothing the outer cortical surface model obtained from the MRI data of normal control #9 in [12] with Weighted-Spherical harmonic representation method which is introduced by Chung et al [14]. The result is the summation of Fourier series on the spatial frequencies of the original surface with a certain low-pass filter. In this simulation, the bandwidth was 0.0001 and the number of iteration was 40 resulting highly smooth surface (**Fig. 9 right**). Recall that this smooth cortex-like surface is used as an example of a manifold with a high complexity like the real cortical surface.

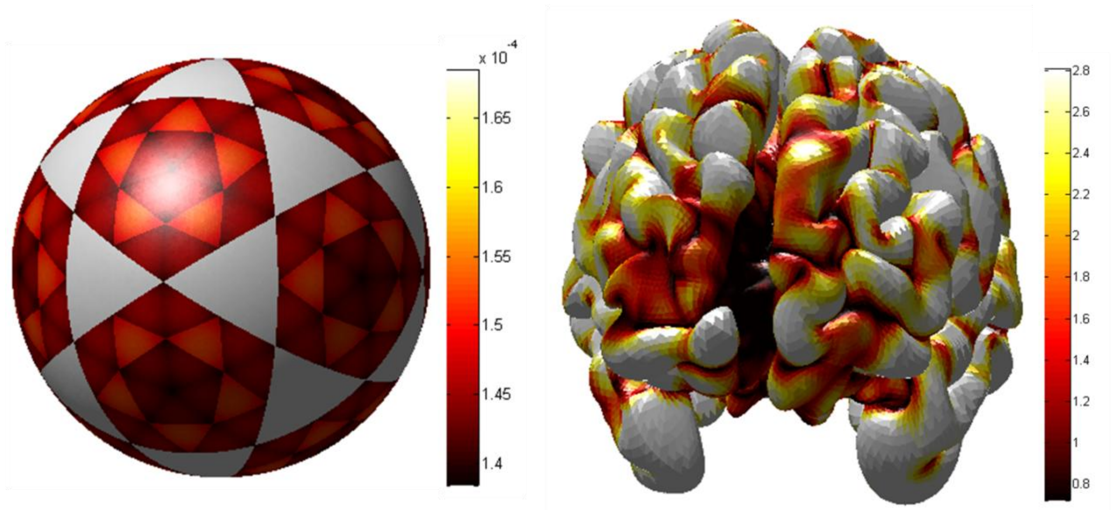

Fig. 9. Smooth surfaces used in simulation; Unit sphere (left) and cortex-like surface (right); Color codes the area of triangles referring to each color bar

Gaussian noise was generated by the pseudo-random number generator built in MATLAB. The standard deviation of the Gaussian distribution with the zero mean was 1/5, 1/4, 1/3 and 1/2 of the mean of edges in the whole mesh respectively. The surface added Gaussian noise (0, mean/2) was visualized for an instance in **Fig. 10**. Each effect of the addition of Gaussian noise is plotted in **Fig. 11** in terms of the areas of triangles in whole meshes. It is shown that as the amount of noise increases the variance of areas also increase and also that the estimated kernel is more skewed. Note that the distribution of the noisy surface with the Gaussian noise (0, mean/2; **Fig. 11**. right column, the lime line) is similar to the one of the unsmoothed raw outer cortical surface (**Fig. 11**. right column, the dotted line) in terms of variance (σ= 1.9185 vs. σ= 2.0505).

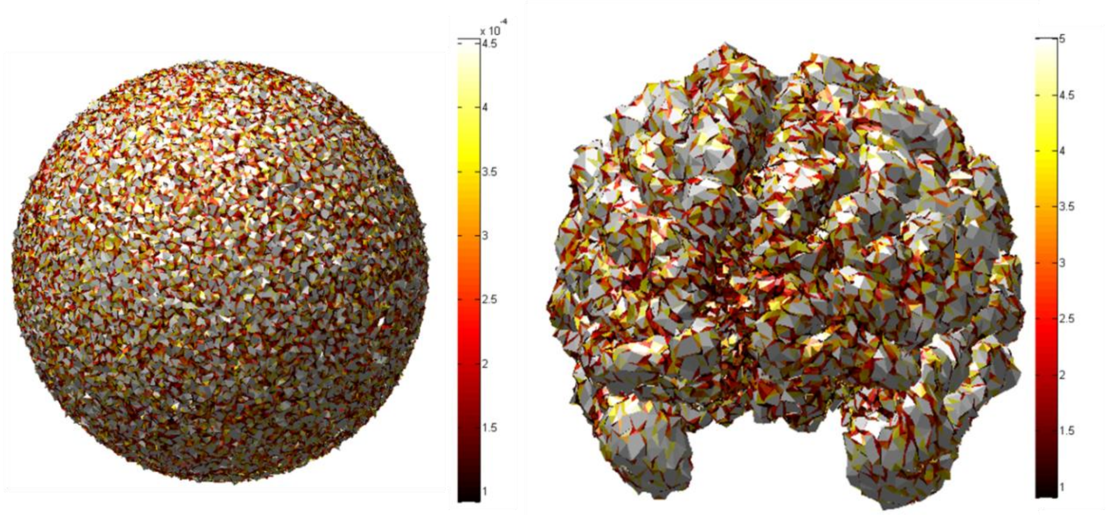

Fig. 10. Noisy surfaces used in simulation; Unit sphere (left) and cortex-like surface (right); Color codes the area of triangles referring to each color bar

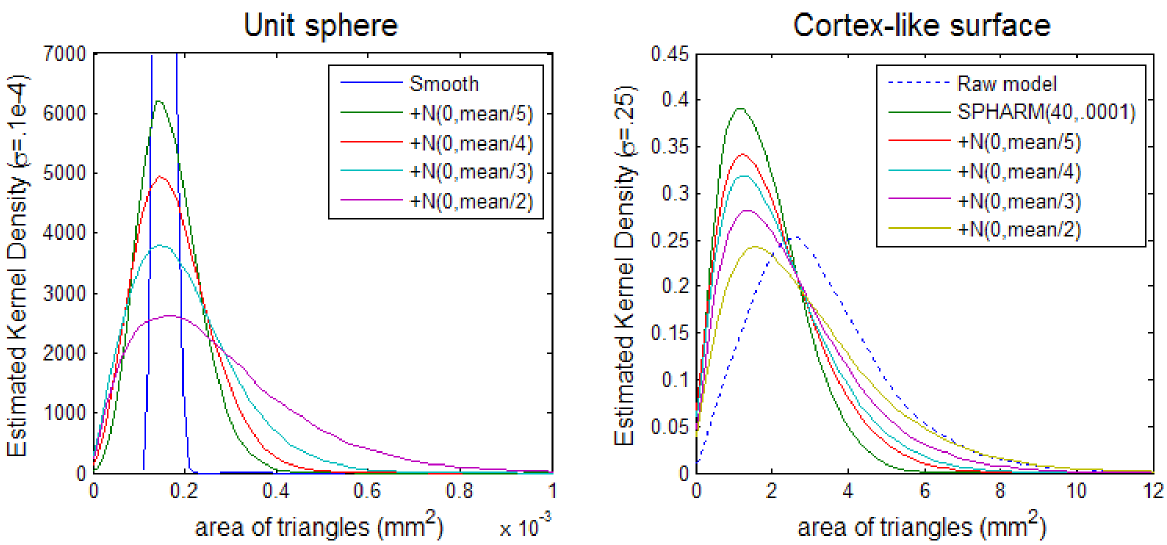

Fig. 11. Effect of Gaussian noise on a unit sphere (left) and a cortex-like surface (right); note that estimated kernel densities are plotted.

3.3. Results

Computed errors from Tabuin's, PCA and Voting methods are plotted (**Fig. 12)**. For both the unit sphere and the cortex-like surface the error of the normal estimation increased as the amount of variance of the Gaussian noise expanded in all three methods. However the means of errors from Voting method (**Fig. 12**; orange lines) increased with the least gradient, and were the least in all cases. Errors from PCA method were the greatest in terms of its value and the degree of increase for all cases and so the standard deviation of errors. The amounts of errors in angle were mapped with colors for both surface models (**Fig. 13**).

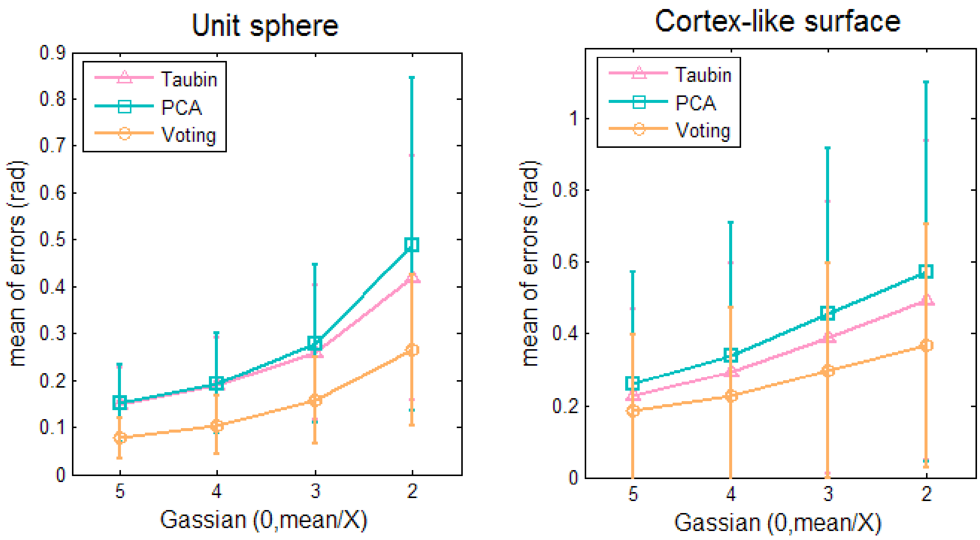

Fig. 12. Amount of noise (mean/X) vs. mean errors from 3 methods; Error bars indicate standard deviations; Horizontal axes are rotated so that the amount of noise increases toward right hand side

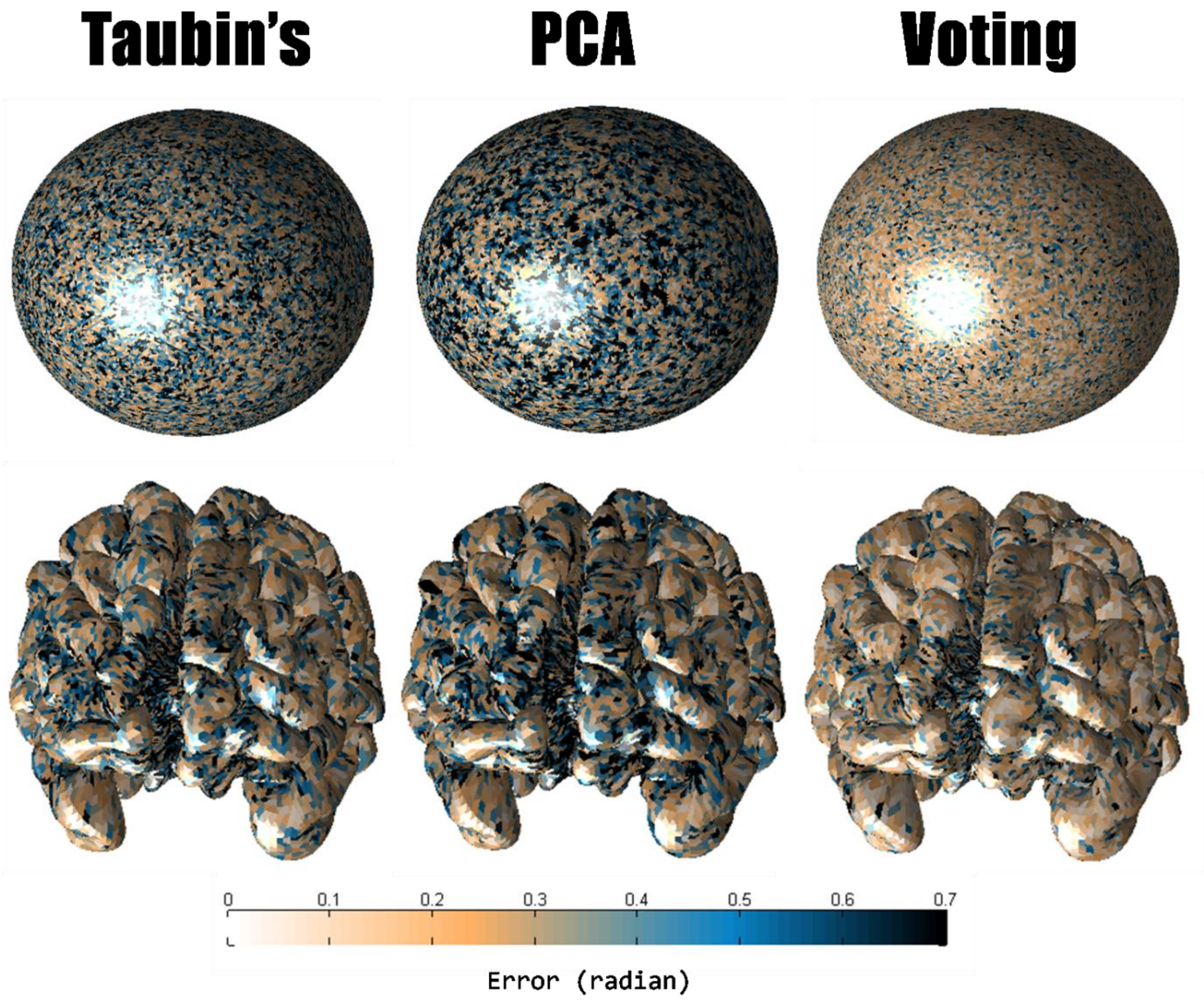

Fig. 13. Error maps from Taubin's method (1st column), PCA method (2nd column) and Voting method (3rd column); Upper row: the unit sphere model; Lower row: the cortex-like surface; Color represents the amount of error on the corresponding vertex in as indicated by the color bar; Lighting effects applied for visualization

4. Discussion

For both the sphere and the cortex-like surface models, the Modified Normal Vector Voting algorithm estimated the normal vectors with the minimum errors among three compared methods. As it is shown in **Fig. 12**, Voting algorithm showed the best performance in all simulated cases, and the improvement of the proposed method in terms of the difference in means of errors enhanced as the amount of noise increased. As assumed, the larger range of neighbors and the penalization of distant neighbor made the estimation less sensitive to the Gaussian noise.

The noise in real surface model in practice is not necessarily Gaussian. Even the similar amount of random noise in terms of variance was generated (**3.2.**), the regional distribution of noise is different from the real data in where the noise is non-uniformly distributed. It is possibly due to not only MRI noise in acquisition but the artifacts from previous processing such as segmentation and surface construction as already mentioned (**1.3.**(2)). Even so, however, the only algorithm can penalize those noise by its distance was Voting method in compared methods. Because of this, the resistance to the noise of Voting method can be well preserved.

The computational load, on the order hand, may be significant cost for Voting method. While Taubin's and PCA methods take less than 2 minutes (96.9131 seconds and 62.5791 seconds respectively) except the previous computation of surface normals on Intel® Core™2 Quad CPU @ 2.33GHz with 3.25GB RAM to estimate 40,962 vertex normals on a single surface, Voting method took more than 20 minutes (1,256.9198 seconds). Though the implementation hasn't been optimized, the high computational load is due to the nature of algorithm including exponentially increased numbers of neighbors. In this implementation, the average number of 1-ring neighbors was 5.99, whereas the average number of 1-ring and 2-ring neighbors was 24.00. In case regarding geodesic distance with larger range of neighbors, the algorithm should be optimized to extend the proposed algorithm to application in practice.

5. References

1. **Page, D.L., et al.,** *Normal vector voting: Crease detection and curvature estimation on large, noisy meshes.* **Graphical Models, 2002. 64(3-4): p. 199-229.**
2. **Weisstein, E.W.** *"Normal Vector." From MathWorld--A Wolfram Web Resource.* **. 2009; Available from: http://mathworld.wolfram.com/NormalVector.html.**
3. **Liu, T., et al.,** *Reconstruction of central cortical surface from brain MRI images: Method and application.* **Neuroimage, 2008. 40(3): p. 991-1002.**
4. **Smith, S.M.,** *Fast robust automated brain extraction.* **Human Brain Mapping, 2002. 17(3): p. 143-155.**
5. **Huang, H., et al.,** *Anatomical Characterization of Human Fetal Brain Development with Diffusion Tensor Magnetic Resonance Imaging.* **J. Neurosci., 2009. 29(13): p. 4263-4273.**
6. **Lin, F.-H., et al.,** *Distributed current estimates using cortical orientation constraints.* **Human Brain Mapping, 2006. 27(1): p. 1-13.**
7. **Gouraud, H.,** *Continuous Shading of Curved Surfaces.* **IEEE Trans. Comput., 1971. 20(6): p. 623-629.**
8. **Jirka, T. and V. Skala,** *Gradient Vector Estimation and Vertex Normal Computation***. 2002.**
9. **Taubin, G.,** *Estimating the tensor of curvature of a surface from a polyhedral approximation.* **Fifth International Conference on Computer Vision, Proceedings.**

1995, Los Alamitos: IEEE, Computer Soc Press. 902-907.
10. Thurmer, G. and C.A. Wuthrich, *Computing vertex normals from polygonal facets.* Journal of Graphic Tools, 1998. 3(1): p. 43-46.
11. Chung, M.K., *Computational Neuroanatomy.* in writing.
12. Chung, M.K., et al., *Cortical thickness analysis in autism with heat kernel smoothing.* Neuroimage, 2005. 25(4): p. 1256-1265.
13. Fischl, B., et al., *High-resolution intersubject averaging and a coordinate system for the cortical surface.* Human Brain Mapping, 1999. 8(4): p. 272-284.
14. Chung, M.K., et al., *Weighted Fourier series representation and its application to quantifying the amount of gray matter.* Ieee Transactions on Medical Imaging, 2007. 26(4): p. 566-581.

Appendix: MATLAB implementation of Modified Normal Vector Voting Estimation

(1) For all triangles in the whole mesh, compute surface normals as the cross product of two edges of triangles. In the i-th row of TRI matrix of MNI format, the indices of vertices constructing the i-th triangle in counterclockwise fashion from the outside. From this information, always outward normals can be computed.

(2) For all vertices in the whole mesh, sequentially define each vertex as the 'center vertex',

A. Find the triangles that contain the center vertex as 1-ring neighbors.
B. Find the triangles that contain the vertices of 1-ring neighbors as 2-ring neighbors as

```
r=[];[r1,c1]=find(tri==center); % 1-ring nbr triangles
for i=1:length(nbr2{center}); % nbr2 cell contains 1-ring nbr vertices id#
       [tr,tc]=find(tri==nbr2{center}(i));
       mr2=[mr2; tr];mc2=[mc2; tc];
end;
    r2=unique(mr2); % 2-ring nbr triangles (including 1-ring nbrs)
```

C. For 1-, 2-ring neighbors,
   i. Rotate the surface normal computed in (1) as a vote of each triangle.
   ii. Compute the L2norm between the center vertex and the centroid of each triangle.
   iii. Product the each area of triangle normalized by the sum of areas of neighbors and the exponential term of the negative L2norm divided by the half of the mean of edges from the whole mesh as a weighting factor.
D. Using the weighting factor above, average all votes from neighbors as an estimation for the center vertex