Toward the Assessment of Intrinsic Geometry of Implicit Brain MRI Manifolds

Karim Makki, Ben Salem, Boulbaba Ben Amor

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xxxxxxxxx

Toward the assessment of intrinsic geometry of implicit brain MRI manifolds

KARIM MAKKI[1,2], (Member, IEEE), DOURAIED BEN SALEM[3], and BOULBABA BEN AMOR[4], (Senior Member, IEEE).

[1]IFP Energies Nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil Malmaison, France
[2]INRIA Rennes Bretagne Atlantique, France
[3]Department of Neuroradiology , CHRU Brest, LaTIM U1101 INSERM, France (e-mail: douraied.bensalem@chu-brest.fr)
[4]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates (e-mail: boulbaba.amor@inceptioniai.org)

Corresponding author: Karim Makki (e-mail: karim.makki@ifpen.fr).

ABSTRACT Principal and Gaussian curvatures are commonly used intrinsic metrics for geometric morphometrics analysis: to assess morphometric changes in brain geometry (developmental neuroimaging studies), to quantify shape deformation (organ motion assessment), or to analyze shape variability across subjects in musculoskeletal studies (statistical shape analysis of bones). However, most of existing algorithms for estimating curvatures act on explicit surfaces (triangle meshes), making them time consuming and sensitive to parameterization and neighborhood size. In this paper, we present a suite of fast and parameterization-free algorithms to estimate second order morphometric parameters given an implicit representation for the surface and without any loss of accuracy. We first show results for direct comparisons of our algorithms with a suite of popular algorithms for estimating curvatures of surfaces represented by triangular meshes. Then, in the context of brain imaging, methods were validated against developmental brain MRI data in which surface based analysis has very often failed. We also provided a modified version of the algorithm that can deal with a Freesurfer output surface mesh, and which was evaluated using an adult brain with more complicated folding patterns. Our algorithm provided a more realistic measures of intrinsic curvature for the white matter (mostly ranged between ± 0.07 mm^{-2}) which confirms its robustness. As compared to mesh-based algorithms, our algorithm reduces computation times from a few minutes to only a few seconds, showing a decrease by a factor of up to 7.

INDEX TERMS Brain morphometry, Gaussian curvature, principal curvatures, implicit surface, parametric surface, fast marching

I. INTRODUCTION

For most of existing algorithms for estimating curvatures of 2D surfaces embedded in 3D space, the surface is given by an explicit form [1] [1], [2]. And the mean and Gaussian curvatures are given by the average (arithmetic mean) and the product (geometric mean) of the two principal curvatures, respectively. One of the major drawbacks of these algorithms is their dependency on mesh quality and neighborhood size. Moreover, they are in general time consuming and may fail to deal effectively with some irregular situations such as singular vertices, irregular vertices, or complicated topologies (*i.e.* surfaces of higher genus), which can prohibit their use in clinical settings. In medical image processing, and more precisely, to study motion, growth, and inter-subject variability in anatomical structures, the common starting point is a ground truth binary segmentation (e.g. of brain [3], organs [4], tumors [5], muscles [6], bones and surrounding tissues [7], etc.). Under such circumstances, implicit surface representations, such as level-sets, offer an alternative solution to directly estimate curvature from binary segmentations [8]–[11], thus allowing the computations to be performed in a cartesian grid. Although the fact that implicit models have already been successfully employed in computer vision applications [12], [13], in particular in medical imaging, see for instance the representation of blood vessel surface implicitly in [14], see also the work of *Mémoli et al.* introducing implicit brain imaging [15], we were surprised to find that a little attention has been devoted to the determination of intrinsic properties of thin brain structures (notably the Gaussian and principal curvatures) without the

[1]Throughout the manuscript, we mean by explicit surface, a triangular surface mesh.

need to parameterize the surface somewhere. Whereas only the divergence formula (extrinsic) is often used in a more general sense, especially for curvature-driven segmentation tasks via the use of the Level set method [16], [17]. And even more recently, a deep learning strategy was designed to estimate only the mean curvature of 2D implicit interfaces [18].

One of the advantages of implicit methods compared to explicit methods is that they can capture fine details describing the cortical folding, thus providing their second order morphometric parameters (maximum and minimum curvatures). Such parameters are the key tools to assess early brain development and to determine relevant biomarkers [19]–[22].

In the same context, and because of the complex geometry of the cortical surface, the cortical thickness was also employed as a morphometric parameter: to establish a biomechanical model of cortical folding patterns [23], or to study neurodegenerative diseases such as Alzheimer [24]. A detailed description of the major limitations and drawbacks of existing parametric methods for determining metrics such as cortical curvature or thickness, in particular for developmental data, is available in [25]. Whereas, cerebral cortex and its reconstruction challenges such as partial volume effect and excessive computation times have been identified in [26].

In this paper, we focus in particular on surface curvature computation and we show the capacity of implicit methods to overcome these limitations by comparing them directly to four popular explicit methods [2], [27]–[29]. In [8], *Goldman* has derived explicit curvature formulas for implicit surfaces and proved their coherence with the classical curvature formulas in differential geometry for parametric surfaces. In [9], the authors showed the performances of such methods in terms of accuracy and computation times. From a practical point of view, a more faster implementation is still required to handle very large data sets and the manner in which the surface is defined need to be more refined regarding its high-impact on the computational accuracy [30]. In this work, we discuss the choice of the implicit scalar function that should be continuously differentiable (smooth) in particular near the shape surface [31].

Furthermore, we show the high-dependency of methods for measuring cortical thickness [32] on the geometry of surrounding tissues. For example, the method of *Yezzi et al.* [32] is commonly used as it can ensure a notion of correspondences to provide thickness values between two non-intersecting boundaries. However, it is not guaranteed that the thin cortical area will be perfectly bounded, and consequently, the estimated thickness values may increase drastically for some cortex regions inwhere both inner and outer boundaries intersect.

To summarize, the main aim of the present study is to provide simple, robust, and fast algorithms which will help the medical imaging community to describe efficiently the geometry of very thin structures in the human body and the study of brain geometry and morphometry represents a good example of its application.

II. CONTRIBUTIONS

The main contributions of this incremental research are:

- In [30], it has been empirically shown that implicit methods apply various boundary conditions, thus affecting the curvature measurement near the boundary, independently of the estimator itself. The curvature estimators proposed in [33] were employed in gray-value volumes. Here, and since we would like to extract curvature information from already segmented volumes, we confine ourselves to the case of binary volumes (segmentations) in order to overcome these limitations. We hereby present a generic implicit function that allows a smooth transition of velocities across the shape boundary, and thereby makes the implicit methods faster, and more accurate than explicit ones especially for very thin structures (see Section III-A).
- As compared to the work of *Mémoli et al.* [15], we pay more attention to the intrinsic surface properties in implicit brain imaging, that have been shown to be more relevant to brain morphometry studies [34].
- We perform multileaved comparisons between implicit and explicit methods, in particular for curvature estimation tasks. Their robustness is evaluated using neonatal brain MRI data[2] from the Developing Human Connectome Project[3] (dHCP) [35], [36].
- We provide an open access code (to estimate curvatures, cortical thickness and sulcal depth), dataset and visualization tools.

III. METHODS

In real-world computer graphics applications, there are many ways to digitally encode geometry. These different ways can be divided into two principal categories: explicit and implicit surface representations [37]. While explicit representations allow shape-encoding by providing a complete information about the location of surface points (*e.g.* pointclouds) and their connectivity (*e.g.* meshes), implicit ones just provide a kind of test to decide whether a point is on the surface or not (*e.g.* Level sets, algebraic surfaces, blobby surfaces, etc.). In short, each category is best suited to a different type of geometry and application. For curvature estimation tasks, it is difficult to decide whether explicit or implicit representation is the best-suited option to encode surface and there is a disagreement about this fundamental choice. In [9], it has been shown that the implicit approaches outperform explicit ones in terms of robustness, accuracy, and runtime to measure curvatures of surfaces. In contrast, *Kronenberger et al.* concluded that mesh-based methods allowed for more accurate curvature estimations, and that the runtimes for both representations were similar for geometries with larger surface densities [30]. In this respect, we investigate more the uncertainty in the representation of the surface and we

[2]Subject recruitment and acquisition protocols were approved by the UK Health Research Authority.

[3]http://www.developingconnectome.org

compare between the two approaches in a more general sense with an application to brain MRI manifolds.

A. IMPLICIT REPRESENTATION OF CURVES AND SURFACES

To represent the shape boundary implicitly, the main idea is to start from the initialization step of the Level set method, commonly used in computational geometry and fluid mechanics for tracking planar curves and curved surfaces [31], [38], [39]. In this setup, the evolving curve/surface is defined by an implicit level-set representation φ, whose spatio-temporal dynamics are governed by an advection partial differential equation (PDE):

$$\frac{\partial \varphi}{\partial t} = v|\nabla \varphi|. \tag{1}$$

where the implicit interface is defined as the isosurface $\varphi(x) = \varphi_0 = cste$, v denotes the interface velocity, $\nabla\varphi$ is the gradient vector field of the scalar function φ, and $|.|$ is the Euclidean norm. If we consider only the static phase (the initialization step), the binary step function (BSF) represents the first trivial example for the initialization of the Level set method. It is simply defined by:

$$\varphi(x) = \begin{cases} c, & if \quad x \in \Omega^+ \\ -c, & otherwise, \end{cases} \tag{2}$$

where c is a positive real constant, $\Omega^+ \subset \Omega$ is the region outside the shape segmentation, and $\Omega \subset \mathbb{R}^n$ stands for the image spatial domain. For $c = 0.5$, such a function is expected to satisfy,

$$|\nabla \varphi| = 1, \tag{3}$$

but exclusively at the shape boundaries $\partial\Omega$. Let us note that step functions are not differentiable in the usual sense, but in the sense of distributions (the derivative of φ is the Dirac delta function $\delta(t)$). Note also that the equation (3) is true only in a general sense [31].

Because of its irregularity, the BSF has been employed to validate the robustness of the methods introduced in [40]. The BSF is simple to implement, but is spatially restricted to the shape boundary and it must be reinitialized as long as the surface evolves with time, similarly to the Heaviside function and its smoothed version [41]. Note that the BSF is just defined to introduce concepts. Otherwise, it cannot be employed in the present work to estimate curvature for implicit surfaces since it is usually contaminated by inaccurate interface normal vectors (*i.e.* it exhibits a non-conservative gradient vector field) [41].

In another study [42], the implicit shape boundary was represented as the zero level set of a signed distance scalar function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$, also known as oriented distance function which can be defined for some distance metric d over Ω by:

$$\varphi(x) = \begin{cases} d(x, \partial\Omega) & if \quad x \in \Omega^+ \\ -d(x, \partial\Omega) & if \quad x \in \Omega^-, \end{cases} \tag{4}$$

where $d(x, \partial\Omega) = \inf_{y \in \partial\Omega} d(x, y)$, $\partial\Omega$ denotes the boundary of Ω, and Ω^- and Ω^+ denote the domains inside and outside the shape, respectively. The metric d defines the way in which the distance will be quantified (*e.g.* the standard Euclidean metric/the scalar product is used to measure the length of the straight-line connecting two points) .In this case, the implicit manifold $\mathcal{M}_0$ is the set of points that belong to the boundary $\partial\Omega$, according to:

$$\mathcal{M}_0 = \partial\Omega = \{x \in \Omega \quad | \quad \varphi(x) = 0\}. \tag{5}$$

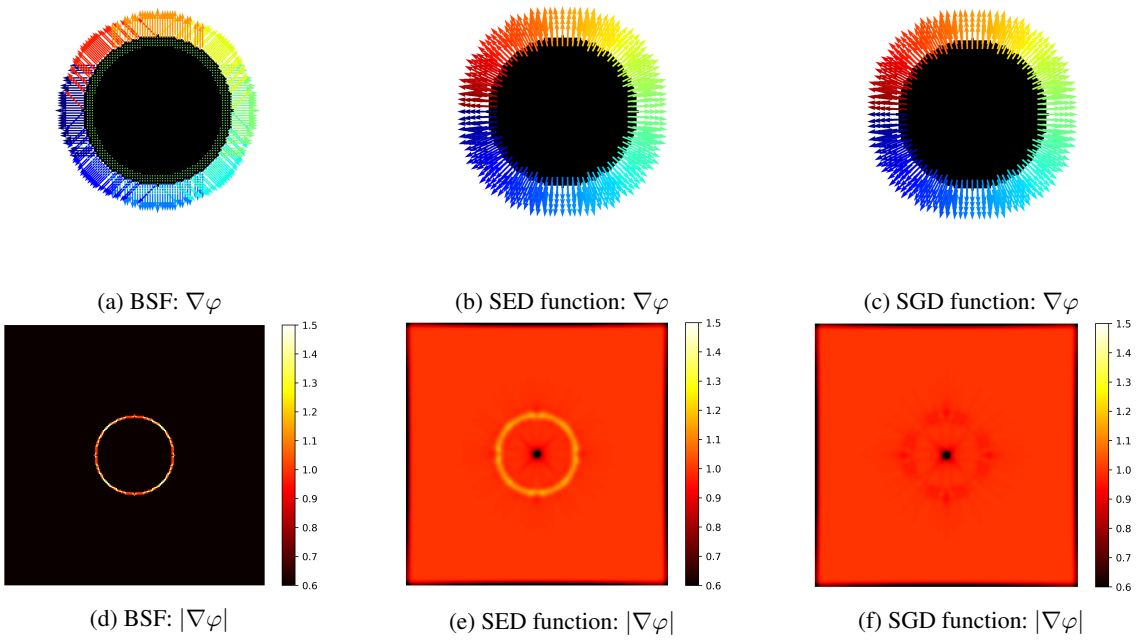
The conventional signed Euclidean distance (SED) function was used to approximate the true distance function in order to provide a complete information about the velocity vector field not only for the interface points, but over the entire spatial domain to establish point correspondences. Which can be thought of as an extension of the velocities defined at the zero level of φ, in the normal direction, to the rest of the domain. While it is clear that the SED function is more flexible and stable than the BSF function, Fig. 1 shows its higher sensitivity to pixelization/voxelization effects. More precisely, the Eq. (3) is not satisfied for points that are equidistant from at least two points on the interface (*i.e.* sharp edges and/or corners). Such problems have been studied in [43], showing also that an approximation of the true distance function may only be continuous but not continuously differentiable because of small errors related to numerical discretization. Assuming that only the zero level set of the function φ is physically meaningful, the fact that the equation (3) may be largely violated, at least in a vicinity of the zero surface, has prevented us from using SED functions for two major reasons:

- The gradient $\nabla\varphi$ and its norm $|\nabla\varphi|$ are implicated in all curvature formulas for implicit surfaces [28]. Therefore, the numerical accuracy and stability of curvature measures will strongly depend on maintaining precision in the gradient of the employed scalar function near the boundary, in terms of both direction and magnitude.
- An extraction of smooth mesh surface (iso-surface, $\varphi(x) = 0$) is required to perform objective comparisons between implicit and explicit methods. This surface extraction can be performed using the marching cubes algorithm [44], for which the computations are also based on the gradient of the scalar field $\nabla\varphi$.

Although the fact that the standard SED functions are much faster to compute, we instead used the fast marching algorithm [45], [46] to estimate maps that we will call, by slightly abusing the notation, signed geodesic distance (SGD) maps for which the property $|\nabla\varphi| = 1$ holds almost everywhere in the image. Thus allowing to avoid the occasional kink where a derivative may fail to exist.

B. SIGNED GEODESIC DISTANCE FUNCTION

As indicated previously, the implicit twice continuously differentiable mapping $\varphi \in \mathcal{C}^2(\Omega)$ is represented by an SGD



(a) BSF: $\nabla\varphi$ (b) SED function: $\nabla\varphi$ (c) SGD function: $\nabla\varphi$

(d) BSF: $|\nabla\varphi|$ (e) SED function: $|\nabla\varphi|$ (f) SGD function: $|\nabla\varphi|$

FIGURE 1: The gradient $\nabla\varphi$ and its magnitude $|\nabla\varphi|$ for the different functions. In the first raw, color indicates the gradient direction in radians on a narrow band of ± 4 pixels around the disk boundary. Colorbar goes from $-\pi$ (blue), to π (red). The second raw shows that the SGD function satisfy the Eikonal equation over the image (in particular, near the boundary and if we expect the central zone characterized by massive vector condensation), which will lead to the most accurate estimates of mesh surface and curvatures, for which, all computations will depend on the gradient itself.

map. The goal is to simultaneously control the direction along which, and the amount by which distance is increasing/decreasing, as we move away from the interface. A nice description of geodesic distance maps and their geometric properties can be founded in [43], [47]. Please keep in mind that the unique difference between what we call SED map and the SGD map is that the value of the scalar function is "explicitly" specified at the boundary for the latter. To estimate φ numerically, we use the fast marching (FM) method to approximate solutions to the stationary boundary value problems of the Eikonal PDE:

$$|\nabla\varphi(x)| = 1/v(x), \quad x \in \Omega \tag{6}$$

subject to the initial Dirichlet boundary condition $\varphi|_{\partial\Omega} = 0$, where Ω is an open set of $\mathbb{R}^n$, and $v(x) > 0$ is the speed on the interface. For a constant unit speed $v(x) = 1$, the solution of this PDE represents an SGD from the boundary $\partial\Omega$ (i.e. the shape surface). One can also interpret the Eikonal equation as a particular case of Eq. (1). Intuitively, for $\frac{\partial\varphi}{\partial t} = 1$, the Eikonal equation simply says that the distance function φ must change with time at a rate of "one millimeter per millimeter." A simple example for a disk of radius $r = 15$ is depicted in Fig. 2.

We show in Fig. 1, for the disk example that the SGD function is characterized by a smooth transition of its gradient from the inside Ω^- to the outside Ω^+ of the shape (and vice versa). It can be thus differentiated across the interface with significantly higher confidence.

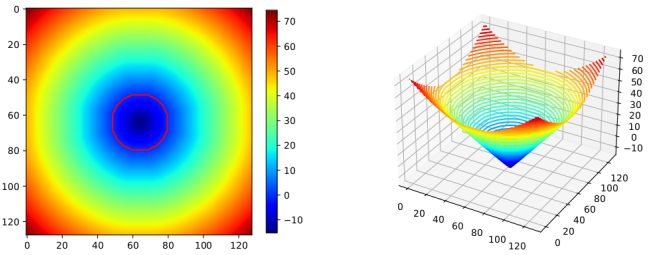


FIGURE 2: 2D example of an SGD function. The theoretical expression of these signed distance functions is: $\varphi_r(x) = |x| - r, \forall x \in \Omega, \forall r \in \mathbb{R}^*_+$.

To speed up computations, a minimum bounding box of the shape point set (mask pixels or voxels) is used to avoid much unuseful computations, in particular, for the image regions very far away from the shape boundaries.

To avoid any kind of confusion, let us emphasize that the FM method was not used to compute shortest path lengths on triangulated surfaces as performed in [35], but rather to compute them for the implicit surface (*i.e.* the Cartesian version) as performed in [15]. Being non-linear (hyperbolic), the Eikonal PDE is just easy to state but not at all easy to

solve. Another advantage of using implicit representations is that the FM algorithm can be easily parallelized for regular grids, but unstructured meshes have proven difficult to handle efficiently.

C. NUMERICAL COMPUTATION OF MEAN CURVATURE

Mean curvature is an extrinsic measure of curvature which corresponds to layman's understanding of curvature before we were ever introduced to differential geometry. Indeed, having the outward unit normal vector field associated to the scalar function φ, $\vec{n}(x) = \frac{\nabla\varphi(x)}{|\nabla\varphi(x)|}$, $\forall x \in \Omega$, the mean curvature can be easily calculated using the divergence formula:

$$K_M = \frac{1}{2}\nabla \cdot (\frac{\nabla\varphi}{|\nabla\varphi|}) = \frac{1}{2} div(\vec{n}) \tag{7}$$

Let us recall that for a *continuously differentiable vector field* defined in Cartesian coordinates by $\vec{n} = n_x i + n_y j + n_z k$, the divergence is defined as the *scalar-valued function*:

$$div(\vec{n}) = \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} + \frac{\partial n_z}{\partial z} \tag{8}$$

To have a smooth unit-length normal vector field along the surface, we perform Gaussian smoothing of $\vec{n}$ in all directions when using the divergence formula. A smoothing of the scalar function φ is also recommended to extract a smooth surface later.

If φ has negative values outside the shape and positive values inside, the mean curvature is simply given by inverting the sign such that $K_M = -\frac{1}{2} div(\vec{n})$, where $\vec{n}$ is the inward unit normal vector field this time. The last equality can be considered as to the "second fundamental form" of an implicit surface.

Although of theoretical interest, *Goldman* showed in [28] that the divergence formula is less practical than other formulas, that depend not only on the trace of the Hessian, but also on its non-diagonal elements. Instead, he proposed the following implicit form of mean curvature:

$$K_M = \frac{1}{2|\nabla\varphi|^3}(\nabla\varphi\nabla^2\varphi\nabla\varphi^T - |\nabla\varphi|^2 trace(\nabla^2\varphi)) \tag{9}$$

The equation above can be also rewritten as:

$$K_M = \frac{1}{2|\nabla\varphi|}(\Delta\varphi - \frac{1}{|\nabla\varphi|^2}\nabla\varphi^T\nabla^2\varphi\nabla\varphi) \tag{10}$$

where $\nabla : \mathbb{R}^n \to \mathbb{R}^n$, $\Delta : \mathbb{R}^n \to \mathbb{R}$, and $\nabla^2 : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ denote the standard differential operators used in *vector/matrix calculus*, namely the gradient, the Laplacian and the Hessian (second-order gradient) operators, respectively. $\nabla\varphi^T$ is the transpose of the gradient vector. Throughout this paper, $|\nabla\varphi|$ and denotes the L^2 norm of the gradient vector. For the signed distance function φ, the Hessian $\nabla^2\varphi$ gives the Weingarten map (shape operator) on the boundary $\partial\Omega$.

Finally, we use the marching squares/cubes to extract the shape surface (iso curve/surface, $\varphi = 0$). In the 3D case, a smooth simplicial mesh (triangular mesh) is generated and a curvature value is associated to each vertex (i.e. the value $K_M(x)$, assuming the coordinates of mesh vertices to be expressed in the image coordinate system). To affect a curvature value to each vertex, we map the real-valued vertex coordinates to new coordinates in the image by nearest neighbour interpolation. Note that any other kind of interpolation can be used here.

D. GAUSSIAN CURVATURE FOR IMPLICIT SURFACES

In differential geometry, the intrinsic curvature at any point of a given surface is proportional to the angle by which a vector rotates when it travels around a local area of the surface, divided by the size of the area as the size of the area approaches zero. In other words, the intrinsic curvature can be defined based on how much a tangent vector rotates when it travels around a local area and then returns to its original position. Gaussian curvature is the major result of Gauss's *Theorema Egregium*. It is a well known intrinsic measure of curvature that does not depend on how the surface is located/embedded in its ambient space. For a point with positive Gaussian curvature, the surface is said to have an elliptic point. For a point with negative Gaussian curvature, the surface is said to have a saddle or hyperbolic point. In [8], *Goldman* has derived Gaussian curvature formulas for implicitly defined curves and surfaces from the classical curvature formulas in differential geometry for parametric surfaces. Based on these formulas, we estimate Gaussian curvature for the implicit surface $\varphi(x) = 0$, by:

$$K_G = \frac{\nabla\varphi\nabla^2_*\varphi\nabla\varphi^T}{|\nabla\varphi|^4} = -\frac{\begin{vmatrix}\nabla^2\varphi & \nabla\varphi^T \\ \nabla\varphi & 0\end{vmatrix}}{|\nabla\varphi|^4} \tag{11}$$

where $\nabla^2_*\varphi$ is the adjoint of the Hessian $\nabla^2\varphi$, such that $(\nabla^2\varphi)_{mn} \equiv \frac{\partial^2 f}{\partial x_m \partial x_n}$, for $m, n \in (i, j, k)^2$. Denoting the 3×3 Hessian matrix by:

$$\nabla^2\varphi = \begin{bmatrix}\varphi_{ii} & \varphi_{ij} & \varphi_{ik} \\ \varphi_{ji} & \varphi_{jj} & \varphi_{jk} \\ \varphi_{ki} & \varphi_{kj} & \varphi_{kk}\end{bmatrix} \quad for \quad x = (i, j, k), \tag{12}$$

the adjoint of $\nabla^2\varphi$ is defined by:

$$\nabla^2_*\varphi = \begin{bmatrix}\varphi_{jj}\varphi_{kk} - \varphi_{jk}\varphi_{kj} & \varphi_{jk}\varphi_{ki} - \varphi_{ji}\varphi_{kk} & \varphi_{ji}\varphi_{kj} - \varphi_{jj}\varphi_{ki} \\ \varphi_{ik}\varphi_{kj} - \varphi_{ij}\varphi_{kk} & \varphi_{ii}\varphi_{kk} - \varphi_{ik}\varphi_{ki} & \varphi_{ij}\varphi_{ki} - \varphi_{ii}\varphi_{kj} \\ \varphi_{ij}\varphi_{jk} - \varphi_{ik}\varphi_{jj} & \varphi_{ji}\varphi_{ik} - \varphi_{ii}\varphi_{jk} & \varphi_{ii}\varphi_{jj} - \varphi_{ij}\varphi_{ji}\end{bmatrix} \tag{13}$$

For more mathematical details and proofs for the definition of $\nabla^2_*\varphi$, the reader is referred to [8].

Once we have the results for mean and Gaussian curvatures K_M and K_G, the principal curvatures k_1 (maximum) and k_2 (minimum) can be deduced by solving the system:

$$\begin{cases} K_M = \frac{1}{2}(k_1 + k_2) \\ K_G = k_1.k_2, \end{cases} \tag{14}$$

which gives:

$$k_1, k_2 = K_M \pm \sqrt{|K_M^2 - K_G|}. \quad (15)$$

These principal curvatures describe the biggest/smallest rate of curvature changes over a surface.

In addition to the explicit formulas for computing mean and Gaussian curvatures for implicitly defined surfaces, *Goldman* proposed an alternative definition for the curvature, called the adjoint Hessian formula which can be directly derived from the above defined Gaussian curvature as follows:

$$\kappa = -|\nabla\varphi|.K_G. \quad (16)$$

Also, for any 2D surface embedded in 3D Euclidean space, the scalar curvature (or the Ricci scalar) can also be determined from the Gaussian curvature by $R = 2K_G$.

E. EIGENVALUES OF THE HESSIAN MATRIX

In the case of parametric surfaces, the second fundamental tensor $\mathbb{II}$ defined in the tangent plane to the surface (also known as the Weingarten map or the shape operator) is given by the following 2×2 real symmetric matrix:

$$\mathbb{II} = \begin{bmatrix} \frac{\partial n}{\partial u} \cdot u & \frac{\partial n}{\partial v} \cdot u \\ \frac{\partial n}{\partial u} \cdot v & \frac{\partial n}{\partial v} \cdot v \end{bmatrix}, \quad (17)$$

where (u, v) are the directions of an orthonormal coordinate system in the tangent frame, and n is the outward-facing normal. The eigenvalues and eigenvectors of $\mathbb{II}$ provide a complete description of surface intrinsic properties such as: principal curvatures (its eigenvalues), Gaussian curvature (its determinant), and the principal directions (its eigenvectors) which are orthogonal in the sense of the scalar product. In contrast, the mean curvature (extrinsic) is proportional to its trace (the average of its eigenvalues). In practice, solving a 2×2 matrix for eigenvalues defined in the tangent plane leads to a quadratic equation. However, the explicit formula for the real symmetric 3×3 matrix is more complicated due to cubic polynomials. And there are no standard geometric (intrinsic) interpretations of the Hessian eigenvalues in any arbitrary basis (i, j, k) if we expect the property telling us that their sum gives the extrinsic Laplacian operator [48] (the divergence formula).

In this work, we start design to investigate experimentally the geometrical properties of the Hessian eigenvalues and some combinations of them when the surface is represented implicitly. To this end, we provide the numerical solutions of the eigenvalue problem $\nabla^2\varphi(v) = \lambda v$:

$$\nabla^2\varphi = P.diag(\lambda_1, \lambda_2, \lambda_3).P^{-1} \quad (18)$$

where the columns of P are the eigenvectors $v_i \in \mathbb{R}^3$, which correspond to the eigenvalues $\{\lambda_i\}_{i\in 1\ldots 3}$ of the Hessian matrix. As detailed in Section IV-E, an exhaustive list of surface geometric features can be derived from these eigenvalues.

F. PROPOSED ALGORITHM

In this section, we summarize the different steps of our algorithm illustrated in Fig. 3. It allows to perform the necessary computations (first and second derivatives) of a scalar function over a regular grid to finally obtain intrinsic curvature and extrinsic features of the surface without having to estimate curvature tensor while being as robust as possible to singularities within the domain.

- To minimize user intervention as much as possible, it is only needed to provide a binary segmentation as input.
- A fast C++ implementation of the FM algorithm is used (*i.e.* we use the scikit-fmm *Python* package which transforms a BSF function into an SGD scalar function with a conservative gradient vector field). Since the complexity of the FM algorithm is $O(nlog(n))$, where n is the number of grid points, we use a bounding box around the shape to reduce the grid size (*i.e.* the cardinal of the domain Ω) and thus to decrease the overall computational complexity (see Section III-B).
- We use Goldman's formulas to estimate both extrinsic and intrinsic curvatures from the smooth SGD map (see Sections III-C, and III-D). We also perform a voxel-wise eigendecomposition of the Hessian matrix of the SGD map to derive extrinsic feature maps in function of its eigenvalues (see Section III-E). Since we compute each feature map overall the domain Ω, the use of a bounding box reduces also the computation times of this crucial step at the cost of changing the image origin which can be easily recovered by a simple coordinate shifting.
- We use classical methods for surface extraction (marching cubes) to obtain a surface mesh which corresponds to the isosurface $\varphi = 0$.
- We preserve only meaningful information from volumetric feature maps by affecting a feature value $\mathcal{T}(x)$ to each mesh vertex x. To address the problem of texture mapping for implicit surfaces, we use two kinds of interpolation: a simple nearest-neighbour interpolation, that significantly reduces the blurring effect by placing a mesh vertex x with (real-valued coordinates) on the grid, such that $x \approx (i, j, k)$, or a nearest-neighbour interpolation, followed by neighborhood-averaging interpolation strategy, which gives a scalar field representing the curvature of each vertex (e.g. 1/6 of the total curvature of all directly-connected grid points.), according to,

$$\begin{aligned}\mathcal{T}(x) = \frac{1}{6}[&\mathcal{T}(i-1,j,k) + \mathcal{T}(i+1,j,k) \\ &+\mathcal{T}(i,j-1,k) + \mathcal{T}(i,j+1,k) \\ &+\mathcal{T}(i,j,k-1) + \mathcal{T}(i,j,k+1)].\end{aligned}$$

IV. RESULTS AND DISCUSSION

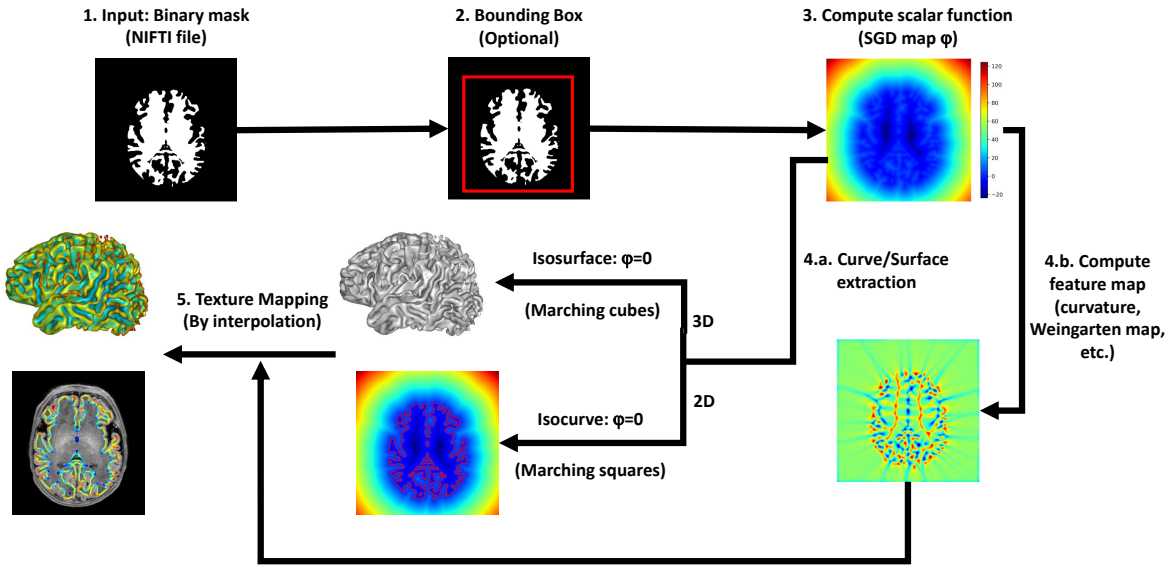


FIGURE 3: Proposed technique for curvature estimation: (1) Input binary segmentation. (2) Define a subimage inside the bounding box surrounding the shape to speed up computations (optional). (3) Compute SGD function using the Fast marching algorithm. (4.a) Exctract isosurface (a triangular surface mesh in the 3D case). (4.b) Estimate a grid feature map (using Goldman's formulas to calculate curvature or matrix eigendecomposition to compute the eigenvalues of the Hessian matrix $\nabla^2\varphi$). (5) Interpolate feature values along the isosurface $\varphi = 0$, extracted using the marching cubes algorithm (per-vertex texture mapping).

A. IMPLEMENTATION TOOLS AND COMPUTATIONAL ASSESSMENT

The NIfTI (Neuroimaging Informatics Technology Initiative) file format is used very commonly in imaging informatics for neuroscience [49], neuroradiology research, and even musculoskeletal [7] and organ motion investigation [4], [50]. In this context, and to avoid duplication of efforts such as mesh post-processing steps (tessellation and refinement), usually required to perform brain feature extraction with Freesurfer (see Section IV-C), we provide an algorithm acting directly on segmented volumes (encoded as NIfTI files), and generating a smooth surface mesh (triangular but not necessarily regular), together with its texture feature maps (per-vertex curvature).

A fast implementation of parameterization-free curvature estimators for manifolds represented as implicit surfaces is then performed in *Python*. The algorithm described in Fig. 3 takes only a few seconds to calculate curvatures for both synthetic and realistic 3D shapes. We also provide tools to display the computed surfaces and features in an object oriented way through the use of the *Visvis Python* library. The source code and data files used in this work are open source and available on GitHub at: https://github.com/k16makki/Medima_tools.

A smooth (twice-continuously differentiable) SGD map is used for surface representation in all experiments which allowed smooth surface mesh extraction using the marching cubes algorithm. It is important to note that a high-quality reconstruction of surface mesh is required to perform objective comparisons between implicit methods and other algorithms for estimating curvature on triangular meshes. We discuss this in more detail in Section IV-C. We further assume that the SGD function should replace the SED function to initialize the dynamical Level set method since the SED may typically develops irregularities, leading thus to serious error propagation during surface evolution.

To construct a texture map for the extracted triangular meshes, we first mapped vertex coordinates to new coordinates in the grid by nearest neighbour interpolation. Second, we associated to each vertex the average texture value of its neighborhood to ensure smooth transitions between concave and convex regions in the surface.

For a variety of shapes including both synthetic that are easy to simulate (*e.g.* deformed sphere, torus, ellipsoid) and realistic (ankle joint manual segmentation from MRI data, neonatal brain), the corresponding mean curvature maps calculated using the equation (9) are presented in Fig. 4, while the corresponding Gaussian curvature maps are depicted in Fig. 5. The example of the the ankle joint complex shows the potential of the algorithm to deal with a set of interactive components (an articulated system) simultaneously. Moreover, the results obtained for a variety of 3D shapes illustrate the robustness of the algorithm under different distributions of triangle sizes and shapes.

The developmental brain MRI data used in this work are from the dHCP project [35]. The white matter and

cortex volumetric segmentations used in the results section are the outputs of the iterative multi-atlas patch-based approach (IMAPA) [51]. Whereas, the hemisphere segmentations shown in Appendix A are generated by the dHCP structural pipeline [4], described in detail in [35]. In contrast, the FLAIR atlas segmentation is available online[5]. For the dHCP subject, the brain second order morphometric parameters (principal curvatures) deduced from mean and Gaussian curvatures according to the equation (15) are provided in Fig. 6.

Having a triangular mesh for which all vertices belong to the zero iso-surface, it would be possible to compare curvatures estimated from implicit surfaces with those produced by existing algorithms acting on triangle meshes [2], [27], [28], [52].

B. COMPARISON BETWEEN IMPLICIT AND EXPLICIT CURVATURE ESTIMATION METHODS

In this section, we perform comparisons between explicit and implicit methods in terms of numerical accuracy, computation times, and robustness, using the surface meshes generated by our algorithm.

Since it is extremely difficult to compare our algorithm to mesh-based curvature estimators in terms of numerical accuracy directly on realistic brain structures, we performed quantitative (numerical) comparisons using a simulated sphere where an analytical solution is available, and qualitative (visual) comparisons using a deformed sphere (for Gaussian curvature) and the Stanford Bunny test model (for mean curvature). For quantitative comparisons, we have employed the different methods to estimate curvatures of a sphere with radius $R = 41$, characterized by a constant mean curvature $K_M = \frac{1}{R}$, and a constant Gaussian curvature $K_G = \frac{1}{R^2}$, in theory. To quantify the vertex-wise differences between the analytical and the estimated curvatures, we have computed the related Root Mean Square (RMS) errors that we have reported in Table 1. The obtained results show clearly that the proposed algorithm outperforms the mesh-based methods in terms of numerical accuracy, at least for this first trivial example. A number of qualitative comparisons of experimental results on curvature in triangular meshes are presented in Figures 7 and 8. Fig. 7 depicts the mean curvature results obtained for the Stanford Bunny (downsampled version, with a vertex count of $\approx 85.5\ k$). Our algorithm is compared to four different methods: the adjacent-normal cubic approximation method (cubic order method) [28], the iterative fitting method [27] (a modified version using only the one ring but including the vertex normals in the fitting process), the Trimesh method for measuring curvature of a sphere centered at a vertex as detailed in [29], and the Rusinkiewicz method for estimating curvature tensor [2]. For the last method, which has been shown to possess a high level of accuracy, in particular for highly resolved regular meshes [23], we employed two different versions of its implementation (both available on GitHub[6] ,[7]).

For both mean (see Fig. 7) and Gaussian (see Fig. 8) curvatures, the performed comparisons show that the proposed algorithm is largely the fastest. In general, the computation time (indicated in each figure caption) decreases by a factor of up to 20 when using a bounding-box around the shape mask. It can often be used to handle large datasets in medical studies. Furthermore, the implicit and explicit curvature methods are numerically and visually distinguishable. Unlike existing mesh-based methods, our algorithm can generate smooth and accurate curvature maps. This suggests that performing the computations in the ambient space leads to less sensitivity to noise and irregularities.

Let us recall that all the comparisons were performed on the surface meshes (simplicial surfaces) extracted using the marching cubes methods, which can ensure topological consistency [53] but do not always generate regular meshes. And that the goal of this section is to evaluate method robustness to poor-quality triangulations. Fig. 9 illustrates for example the robustness of implicit methods in the face of such complex irregular vertices for the torus example (genus-1), which is just a natural consequence of their full non-parametric nature. On the other hand, the robustness of the different methods is evaluated using a realistic neonatal brain segmentation. Despite being able to deal with synthetic surfaces at different levels of precision, all the "explicit" algorithms listed previously failed to estimate curvatures for the cortex and the white matter. More experiments and discussions about the limitations of explicit methods are provided in Appendix B.

C. CURVATURE ESTIMATION FOR FREESURFER OUTPUT: AN ADULT BRAIN CASE

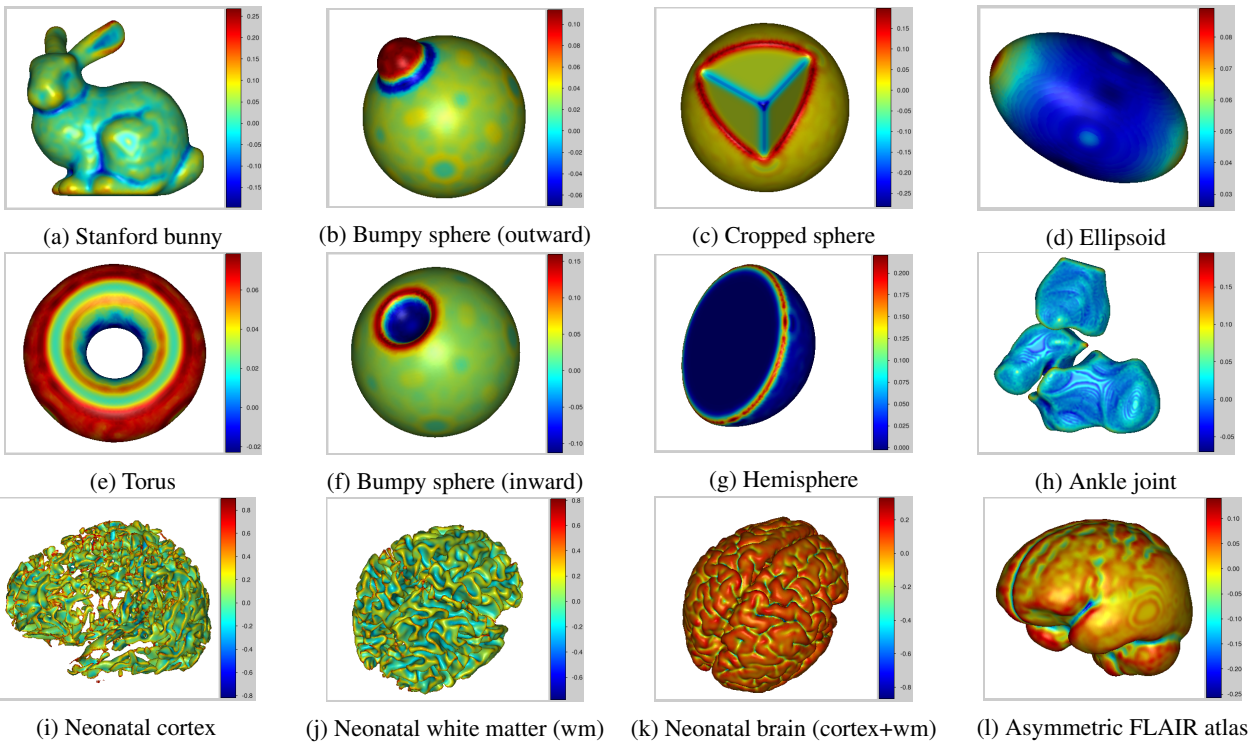
As mentioned in Section IV-A, the marching cubes algorithm generates a topologically correct meshes, but without guaranteeing their regularity which reduces the quality of geometric approximation for mesh-based algorithms. To address this issue, we modified our algorithm (used for method comparisons in Section IV-B) inorder to compare it with mesh-based curvature methods on a quasi-regular mesh generated by Freesurfer. A second aim of this experiment is to evaluate the flexibility of the proposed algorithm on a mesh which was not directly extracted from the scalar signed distance function from which the feature maps were obtained.

FreeSurfer is an open source software platform providing powerful tools which are commonly used in neuroimaging to process and visualize structural and functional neuroimaging data from cross-sectional or longitudinal studies [55]–[57]. It provides methods for the construction of cortical surface models by tessellating topologically correct white matter tissue segmentations. As illustrated in Fig. 10a, a Freesurfer output surface consists of a 3D triangulated mesh placed at

[4] https://github.com/BioMedIA/dhcp-structural-pipeline
[5] https://figshare.com/s/13926af9a272fe26eeb0
[6] https://github.com/brain-slam/slam
[7] https://github.com/AbhilashReddyM/curvpack

TABLE 1: RMS errors associated with curvature estimation using the different methods: sphere example.

Methods	Cubic order	Iterative fitting	Rusinkiewicz	Trimesh	Proposed
Gaussian curvature	$13 * 10^{-3}$	0.145	$6 * 10^{-3}$	$7 * 10^{-3}$	$1.4 * 10^{-3}$
Mean curvature	0.06	0.05	$22 * 10^{-3}$	$18 * 10^{-3}$	$15 * 10^{-3}$



(a) Stanford bunny (b) Bumpy sphere (outward) (c) Cropped sphere (d) Ellipsoid

(e) Torus (f) Bumpy sphere (inward) (g) Hemisphere (h) Ankle joint

(i) Neonatal cortex (j) Neonatal white matter (wm) (k) Neonatal brain (cortex+wm) (l) Asymmetric FLAIR atlas

FIGURE 4: Mean curvature results for synthetic and realistic 3D shapes.

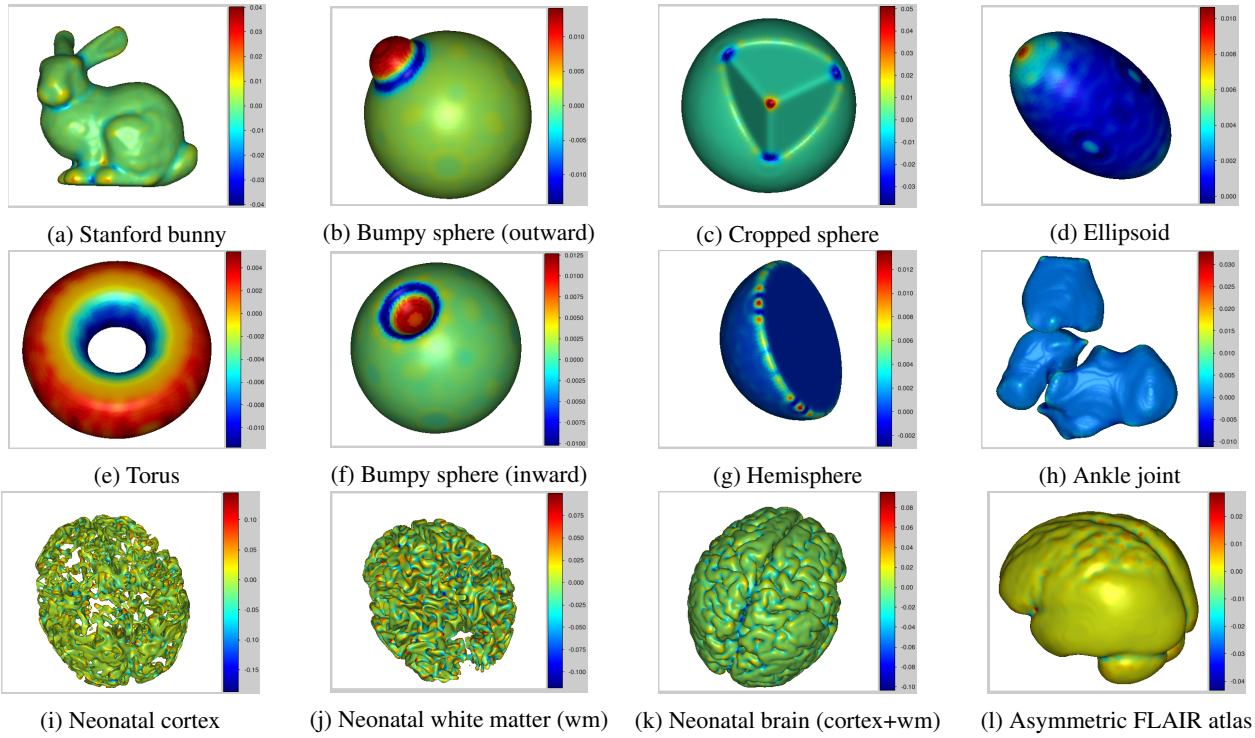
the boundary of the gray and white matter of the brain. This refined mesh is then used to assist with estimating curvatures on the boundary using the different methods. Experimentally, we perform our comparisons using a Freesurfer segmented output data for a human adult brain[8] (a 3D isotropic 1mm^3 MRI volume). This choice was motivated by the fact that mature brains have more complicated folding patterns, as compared to the neonatal brain. Such an increase in the degree of cortical folding would increase the complexity of curvature map to be estimated and its variability across the interface.

Three methods were used to calculate extrinsic/intrinsic curvature per vertex of the right hemisphere surface (a locally refined triangle mesh composed of a set of $n = 144\mathbf{k}$ vertices). The corresponding results are presented in Fig. 10. In this figure, we show that implicit methods can deal more robustly with Freesurfer outputs. Our algorithm provided a more realistic measures of curvatures for the white matter: the intrinsic curvature mostly ranged between -0.07 and 0.07 mm^{-2} over the surface, while the estimated mean curvature ranged over $[-0.5, 0.6]\text{mm}^{-1}$ in average. In contrast, popular mesh-based methods such as the Rusinkiewicz method which was recently used in [23], and the Trimesh method, failed to deal with some irregular vertices, leading to an undesirable out of range of curvature values.

Using our algorithm, the computation times for Gaussian curvature estimation were reduced from 345s to estimate curvature tensors (Rusinkiewicz method) to only 50s (without using a bounding box window). The considerable differences in computational times point to the efficiency of the proposed algorithm.

Note that the proposed algorithm requires both the binary mask (NIfTI file) and the the triangular mesh (GIFTI object: the surface-file format complement to the NIfTI volume-file) as inputs in this new setting. The NIfTI array was used to perform numerical computations of curvature on a $256 \times 256 \times 256$ Cartesian grid and the NifTI header was used to map array coordinates to vertex coordinates expressed in subject-centered scanner coordinates (using the inverse of the NIfTI affine matrix as a transformation for texture mapping between spaces). A modified version of the algorithm described in Fig. 3 is then used, as illustrated in

[8]Subject was selected from the Open Access Series of Imaging Studies (OASIS) database (www.oasis-brains.org). Subject participated in accordance with guidelines of the Washington University Human Studies Committee.

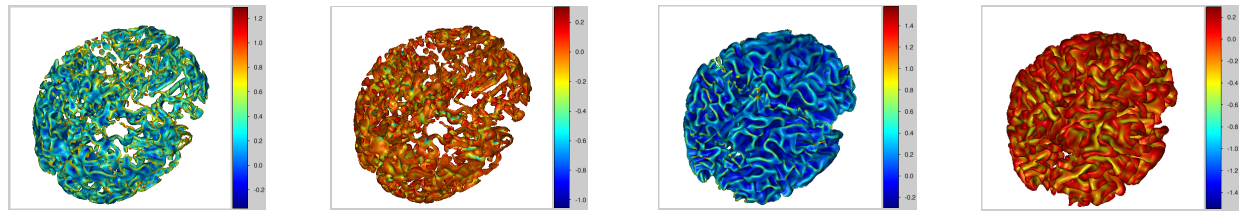


(a) Stanford bunny (b) Bumpy sphere (outward) (c) Cropped sphere (d) Ellipsoid

(e) Torus (f) Bumpy sphere (inward) (g) Hemisphere (h) Ankle joint

(i) Neonatal cortex (j) Neonatal white matter (wm) (k) Neonatal brain (cortex+wm) (l) Asymmetric FLAIR atlas

FIGURE 5: Gaussian curvature results for synthetic and realistic 3D shapes.



(a) k_1 (b) k_2 (c) k_1 (d) k_2

FIGURE 6: Principal curvatures (maximum k_1) and (minimum k_2) for the cortex (a and b), and for the white matter (c and d).

Algorithm 1. The limitation in the present work is that we are considering only isotropic metrics for now.

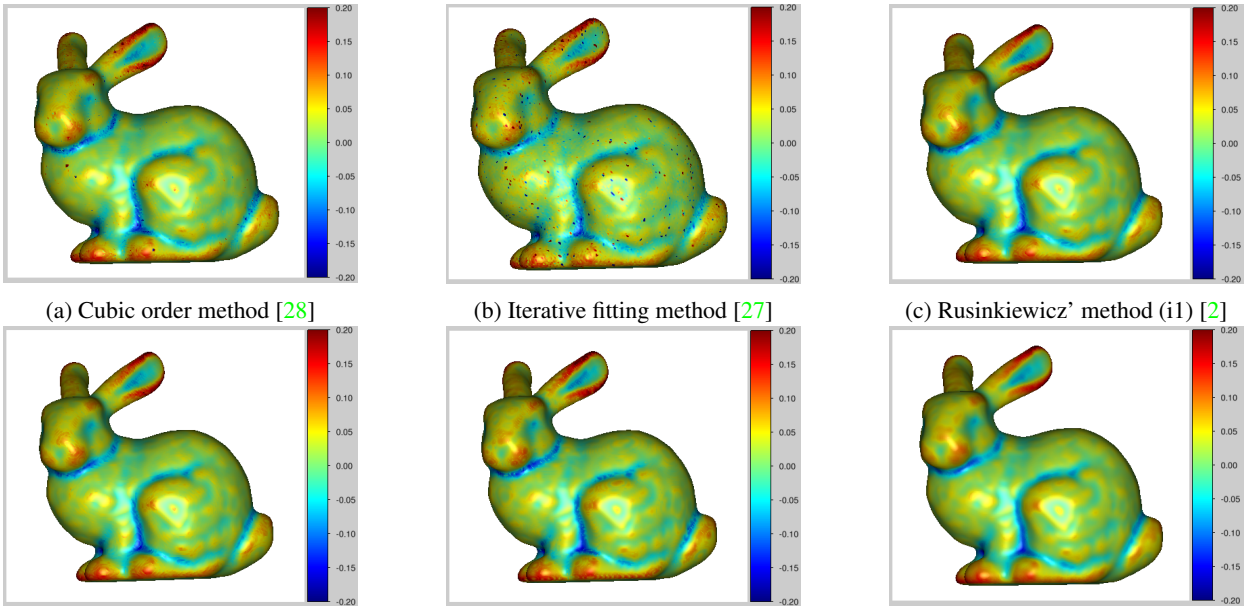
These results demonstrate improved quality of the geometric approximation when using our algorithm, which will increase the possibility of excellent integration of these tools in Freesurfer.

D. CAPACITY TO DEAL WITH THIN STRUCTURES

One of the advantages of the proposed algorithm is that it can be used to characterize the geometry of very thin structures regardless of the complexity of their topologies. In this section, we briefly evaluate the efficacy of the proposed methods to study the geometry of the neonatal cortex. For the neonatal brain, much of the interior volume is occupied by white matter, but the neurons of the cortex only reside in a thin layer of gray matter (only 2-4 mm thick). This makes it difficult to deal with the cortex as an independent structure and to characterize the evolution of its own geometry over a long

Algorithm 1 Estimate curvature for a Freesurfer output surface

- **Inputs:** - FS output surface mesh (GIfTI object)
  - Corresponding NIfTI volume file (binary mask)
- **Curvature estimation:**
  - Compute SGD map φ for the binary mask
  - Compute mean and Gaussian curvatures using Goldman's formulas (grid feature map)
- **Texture mapping between spaces:**
  - Establish correspondences between mesh vertex coordinates expressed in scanner coordinate system and their locations expressed in image coordinate system (real-valued coordinates)
  - Affect a curvature value per vertex by interpolation on the grid feature map (*e.g.* nearest neighbour)
- **Output:** Per-vertex curvature vectors.



(a) Cubic order method [28] (b) Iterative fitting method [27] (c) Rusinkiewicz' method (i1) [2]

(d) Rusinkiewicz' method (i2) (e) Trimesh method [29] (f) Our algorithm (implicit)

FIGURE 7: Mean curvature estimation using different methods (implicit VS explicit). In terms of computation times (provided in seconds below), a modified version of the cubic order algorithm takes 31s; the iterative fitting method takes 383s; the first implementation (i1) of the Rusinkiewicz algorithm takes 120s; the second implementation (i2) of the same algorithm takes 80s; the Trimesh method takes 140s, while our algorithm takes less than 6s. To make the different methods visually comparable, all results are shown with a unique (truncated) colorbar.

period of time. To overcome these limitations, the curvature is often estimated at the interface delineating between the gray and white matters. In the same context, some other feature maps have been employed such as cortical thickness and sulcal depth (see Appendix A).

Concerning the cortical thickness, Fig. 11 shows that the technique proposed in [32] is very performant but can fail in some situations since it provides accurate thickness measures as long as the two boundaries do not intersect. A condition which is not always directly available as illustrated in Fig. 11.a. The results presented in Figures 4.i, 6, and 12.a show that we can deal with the cortex as an independent structure while providing a very useful description of its geometry via mean and Gaussian curvatures, regardless of whether the geometry of its neighbouring structures is complex or not. Based on these encouraging results, we intend to quantify local cortical gyrification (*i.e.* the process of forming the characteristic folds of the cerebral cortex) using the estimated curvature maps as morphometrics.

E. PERSPECTIVE

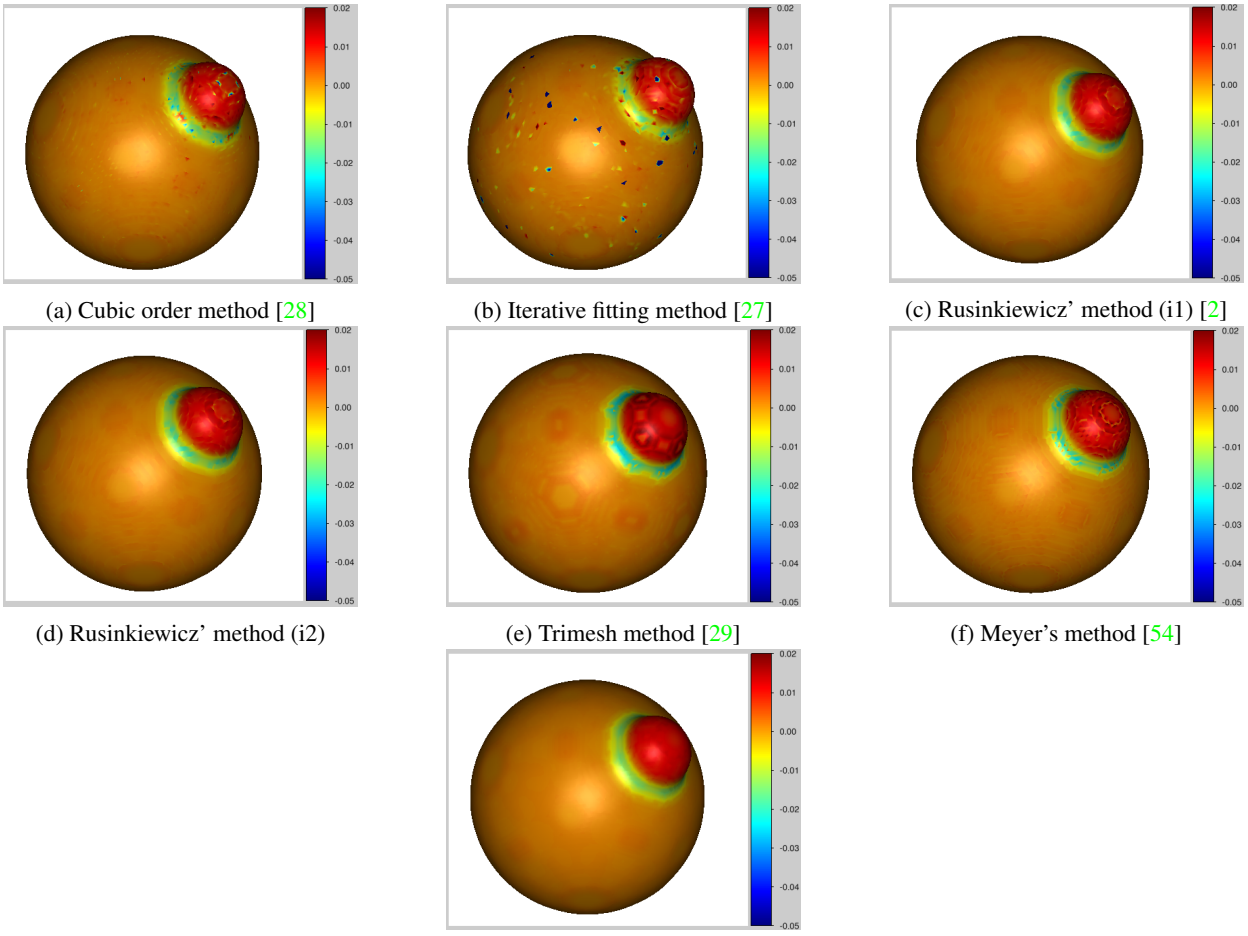
The numerical results presented in Fig. 13 show clearly that the three eigenvalues of the Hessian of the SGD function have nice geometric properties which open the question if there exist some combinations of them that may give the same intrinsic properties as those provided by the shape operator in the case of parametric surfaces. Of course in any arbitrary basis (i, j, k) and without having to express the curvature tensor in the normal frame as proposed in [58], and employed in [9] to derivate formulas for principal directions of implicit surfaces. To deduce the intrinsic geometrical properties of the implicit surface from the Hessian eigenvalues and eigenvectors, a good perspective would seek to learn the mappings relating the two principal curvatures k_1, k_2 and directions t_1, t_2, to the eigenvalues λ_i and eigenvectors v_i of the 3×3 Hessian matrix, respectively. In other words, using a wide variety of shapes as training data sets, we suggest to learn the maps f, F, g, and G, such that:

$$\begin{cases} k_1 = f(\lambda_1, \lambda_2, \lambda_3) \\ k_2 = g(\lambda_1, \lambda_2, \lambda_3) \\ t_1 = F(v_1, v_2, v_3) \\ t_2 = G(v_1, v_2, v_3). \end{cases} \tag{19}$$

with the help of the prior information (divergence formula) that we have at hand: $k_1 + k_2 = \lambda_1 + \lambda_2 + \lambda_3$.

V. CONCLUSION

In this work, we conducted direct comparisons between implicit and explicit approaches for characterizing brain intrinsic geometry that we believe have not yet appeared in the literature. We show that rigorously defining the scalar function to implicitly represent shape surfaces allows to improve robustness, and to reduce computation times for numerically estimating both their extrinsic and intrinsic curvatures. Since it is difficult to compare between implicit methods and the



(a) Cubic order method [28]

(b) Iterative fitting method [27]

(c) Rusinkiewicz' method (i1) [2]

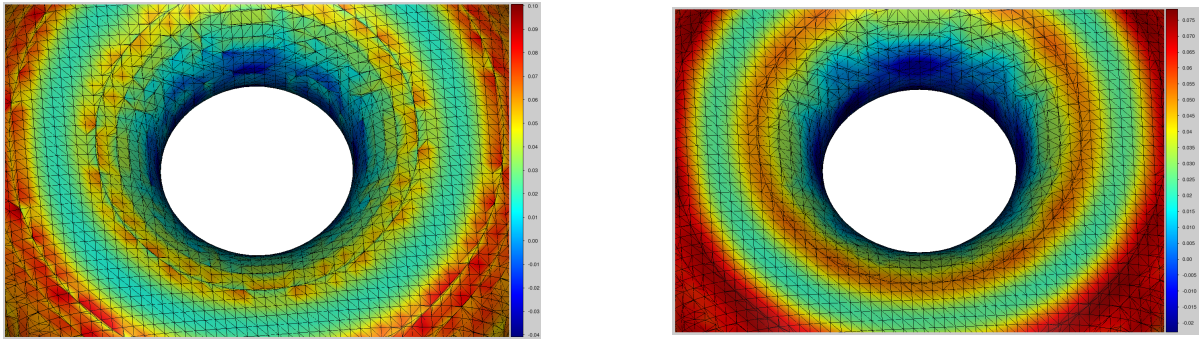
(d) Rusinkiewicz' method (i2)

(e) Trimesh method [29]

(f) Meyer's method [54]

(g) Our algorithm

FIGURE 8: Gaussian curvature estimation using different methods. In terms of computation times (in seconds), a modified version of the cubic order algorithm takes 5.7s; the iterative fitting method takes 80s; the first implementation (i1) of the Rusinkiewicz algorithm takes 27s; the second implementation (i2) of the same algorithm takes 15s; the Trimesh method takes 0.2s; the method of Meyer takes 94s; and our algorithm takes 0.5s. Results are also shown with a unique colorbar (truncated).

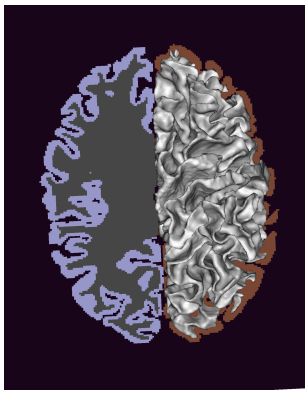


(a) The algorithm described in [2]

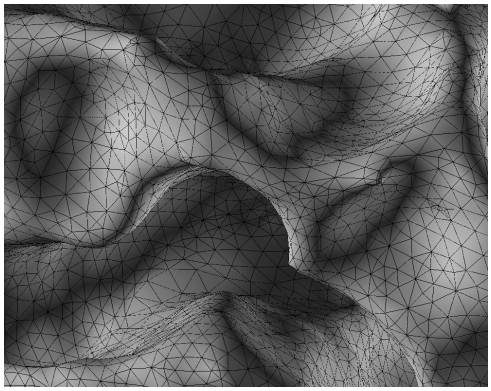
(b) Our algorithm

FIGURE 9: Sensitivity to irregular vertices.

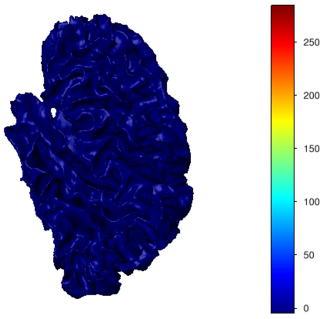
most popular mesh-based methods for estimating curvatures of brain structures in terms of accuracy, the accuracy of each approach is evaluated both qualitatively and quantitatively against artificial data. The obtained results for realistic brain surfaces show also that implicit methods are much more robust and less time-consuming than methods acting on surface



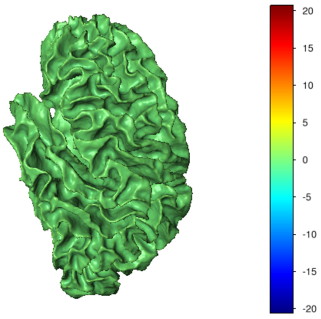
(a) Right hemisphere surface



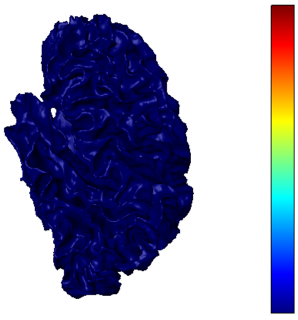
(b) Mesh quality



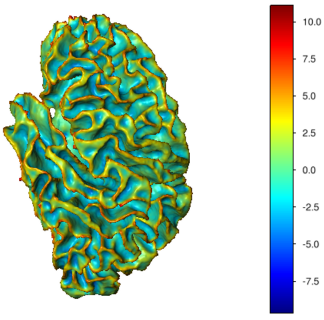
(c) Gaussian curvature (Rusinkiewicz)



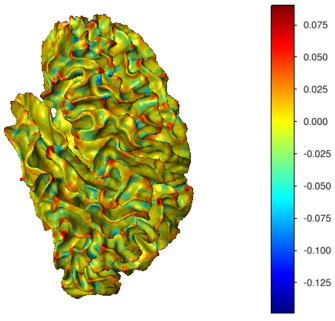
(d) Mean curvature (Rusinkiewicz)



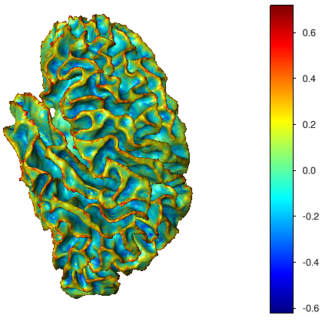
(e) Gaussian curvature (Trimesh)



(f) Mean curvature (Trimesh)



(g) Gaussian curvature (our algorithm)

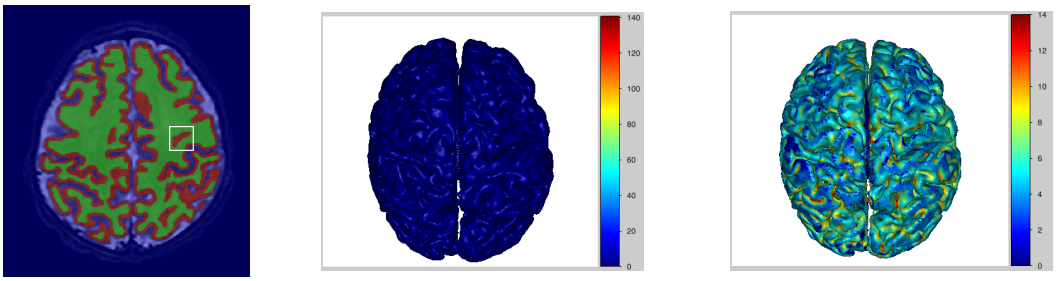


(h) Mean curvature (our algorithm)

FIGURE 10: Curvature results (explicit Versus implicit methods): for the right hemisphere surface mesh generated by FreeSurfer for an adult brain.

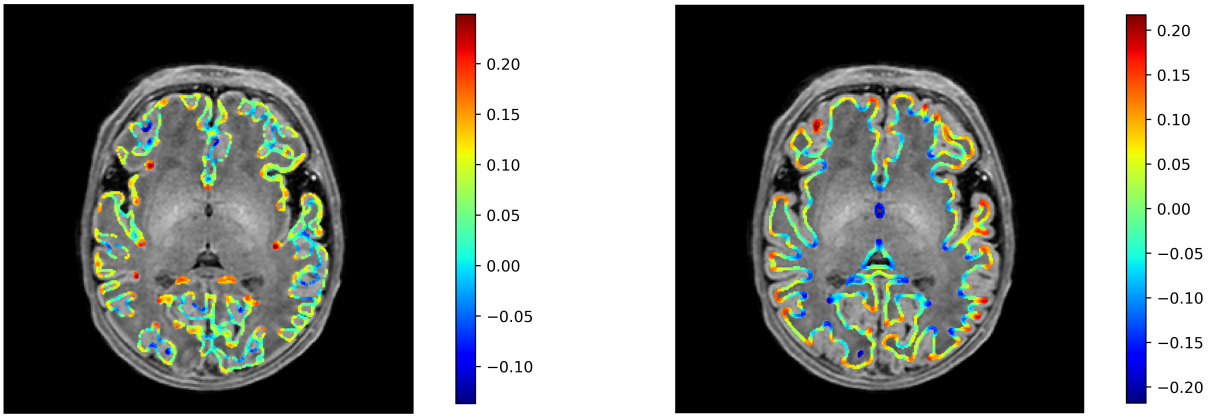
.

meshes which were essentially sensitive to vertex irregularity. In future works, the proposed tools will be used to quantify sulcal, and gyral development of the brains of infants using a large dataset.



(a) (b) (c)

FIGURE 11: Cortical thickness (in mm) using the method described in [32]: (a) brain anatomy and initial boundaries, white matter (inner boundary) in green, cortex segmentation in red, and outer boundary (the rest of the image space) in blue; (b) cortical thickness without imposing constraints on thickness values; and (c) cortical thickness map after imposing a maximal thickness value of 14 mm.



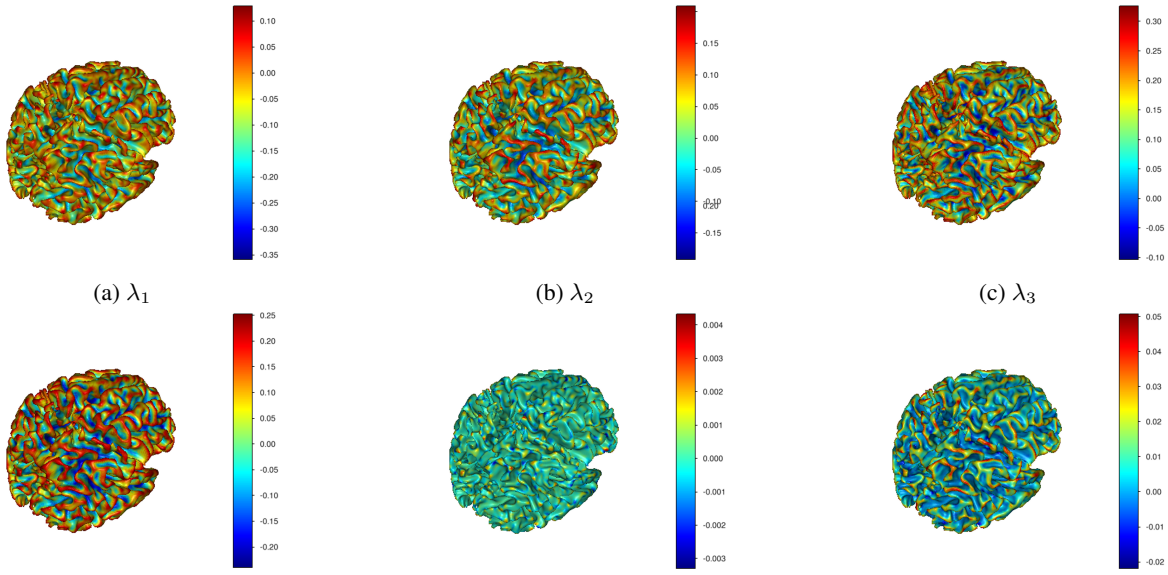
(a) Cerebral cortex (b) White matter

FIGURE 12: Per-slice mean curvature: anatomical T1-weighted neonatal MRI slice and estimated 2D mean curvature for cortex and white matter, respectively.

.

APPENDIX A SULCAL DEPTH MAPS

In addition to the surface curvature and cortical thickness, sulcal depth maps are also of of great interest to the neuroimaging community as they provide information about folding amplitude and dynamic patterns of cortical expansion during folding [35], [59]. In [35], the sulcal depth was defined as the average convexity or concavity of cortical surface points. In [60], the sulcal depth map was defined as the shortest paths from the 3D convex hull to the vertices of the cortical surface. And the shortest paths were computed using the Dijkstra algorithm. A major drawback of the Dijkstra algorithm is that the direction along which distance increases, is partially ignored (*i.e.* only vertical and horizontal displacements are allowed). One can imagine, for instance, that this algorithm will overestimate the straight-line Euclidean distance of any diagonal path, crossing a regular grid. To surmount this issue in the suite of codes that we share, we provided an adapted version of the algorithm originally proposed in [32] for the estimation of tissue thickness, to estimate sulcal depth maps. The second definition of sulcal depth maps is considered and a notion of anatomical correspondences is developed. In fact, the latter method provides a way to split the non-linear shortest path problem into two linear (elliptic) sub-problems that are easy to solve subsequently. It involves three principal steps: the first one consists of solving a two-point Dirichlet boundary value problem. More explicitly, this step aims at finding the harmonic scalar function u, which takes prescribed values at the boundaries of a region Ω ($u(\partial_0\Omega) = 1$ and $u(\partial_1\Omega) = 0$) while satisfying the elliptic Laplace PDE $\Delta u = 0$ in-betweens (inside Ω). The idea behind is to specify the direction along which the distance will increase assuming that the gradient vector field ∇u is parallel to the gradient vector field of the true scalar distance function we are looking for. The second step consists of a simple change of variable by computing $N = -\frac{\nabla u}{||\nabla u||}$ in order to control the magnitude (the amount by which) distance will increase and thus to indirectly satisfy the Eikonal PDE inside Ω thereafter. And



(a) λ_1 (b) λ_2 (c) λ_3

(d) $\frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_3)$ (e) $\lambda_1.\lambda_2.\lambda_3$ (f) $\lambda_2.\lambda_3$

FIGURE 13: Eigenvalues $\{\lambda_i\}_{i\in 1\ldots 3}$ of the Hessian of the SGD function and some of their combinations on the shape boundary. The eigenvalues are sorted pointwise into ascending order: $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

the third step consists of recovering the true distance function $\phi = L_0 + L_1$ from N, by minimizing $\sum_{n=0}^{1} \int_\Omega |\nabla L_n - N|^2$, or equivalently, by solving the linear (elliptic) Poisson PDE: $\Delta\phi = \nabla \cdot N$. This last step aims at finding the geodesic distance map whose gradient has unit length and points in the right direction. For more details about these briefly described steps, the reader is referred to the original paper [32]. An example of its application to the neonatal brain is illustrated in Fig. 14a.

APPENDIX B LIMITATIONS OF EXPLICIT METHODS

In this Appendix, we present the obtained curvature maps for the right hemisphere of the dHCP subject, generated by the dHCP structural pipeline. Despite the high quality of the input mesh, the obtained results were unsatisfactory (see Fig. 15). For most of explicit methods, a biased estimation was manifested by an out-of-range of the curvature values. This problem was caused by the presence of some irregular vertices, at which the curvature increases/decreases drastically. Note that this problem can be solved by truncating the range of values, but this requires a large amount of manual intervention. Note also that all these methods were very time-consuming.

REFERENCES

[1] C.-s. Dong and G.-z. Wang, "Curvatures estimation on triangular mesh," Journal of Zhejiang University-Science A, vol. 6, no. 1, pp. 128–136, 2005.

[2] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. IEEE, 2004, pp. 486–493.

[3] T. D. Bui, L. Wang, J. Chen, W. Lin, G. Li, and D. Shen, "Multi-task learning for neonatal brain segmentation using 3D dense-unet with dense attention guided by geodesic distance," in Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data. Springer Nature Switzerland, 2019, pp. 243–251.

[4] K. Makki, A. Bohi, A. C. Ogier, and M. E. Bellemare, "A new geodesic-based feature for characterization of 3D shapes: application to soft tissue organ temporal deformations," in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 3822–3828.

[5] S. Tan, L. Li, W. Choi, M. K. Kang, W. D D'Souza, and W. Lu, "Adaptive region-growing with maximum curvature strategy for tumor segmentation in 18F-FDG PET," Physics in Medicine & Biology, vol. 62, no. 13, p. 5383, 2017.

[6] P.-H. Conze, S. Brochard, V. Burdin, F. T. Sheehan, and C. Pons, "Healthy versus pathological learning transferability in shoulder muscle MRI segmentation using deep convolutional encoder-decoders," Computerized Medical Imaging and Graphics, vol. 83, p. 101733, 2020.

[7] K. Makki, B. Borotikar, M. Garetier, S. Brochard, D. B. Salem, and F. Rousseau, "In vivo ankle joint kinematics from dynamic magnetic resonance imaging using a registration-based framework," Journal of biomechanics, vol. 86, pp. 193–203, 2019.

[8] R. Goldman, "Curvature formulas for implicit curves and surfaces," Computer Aided Geometric Design, vol. 22, no. 7, pp. 632–658, 2005.

[9] E. Albin, R. Knikker, S. Xin, C. O. Paschereit, and Y. d'Angelo, "Computational assessment of curvatures and principal directions of implicit surfaces from 3D scalar data," in International Conference on Mathematical Methods for Curves and Surfaces. Springer International Publishing, 2016, pp. 1–22.

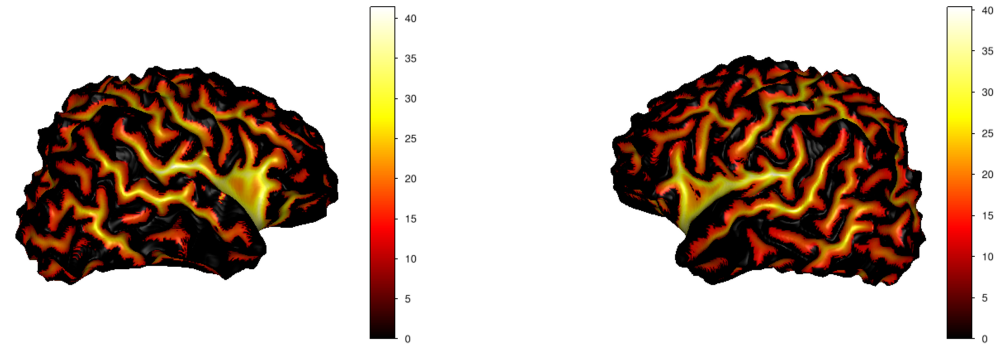
[10] C. Bermudez, A. J. Plassard, L. T. Davis, A. T. Newton, S. M. Resnick, and B. A. Landman, "Learning implicit brain MRI manifolds with deep learning," in Medical Imaging 2018: Image Processing, vol. 10574. International Society for Optics and Photonics, 2018, p. 105741L.

[11] A. Song, "Generation of tubular and membranous shape textures with curvature functionals," arXiv preprint arXiv:2103.04856, 2021.

[12] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

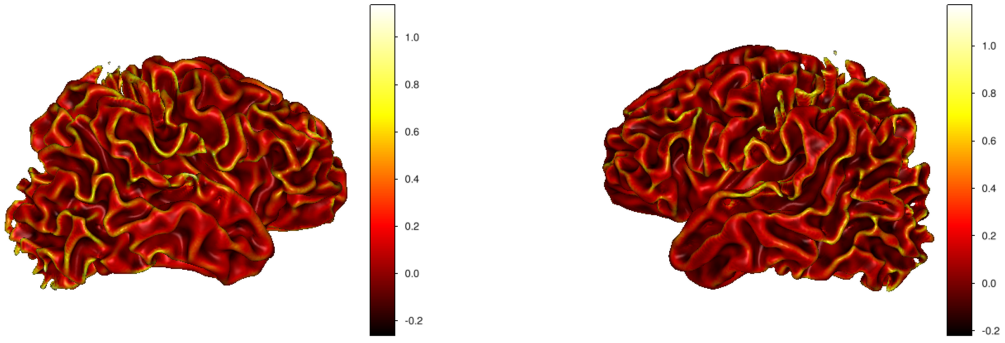
[13] O. Poursaeed, M. Fisher, N. Aigerman, and V. G. Kim, "Coupling explicit and implicit surface representations for generative 3D modeling," in European Conference on Computer Vision. Springer International Publishing, 2020, pp. 667–683.

[14] E. Kerrien, A. Yureidini, J. Dequidt, C. Duriez, R. Anxionnat, and S. Cotin, "Blood vessel modeling for interactive simulation of interventional neu-



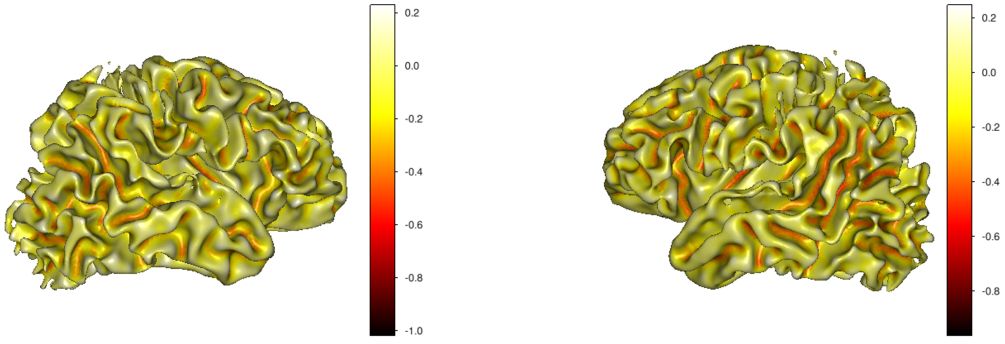
(a) Sulcal depth map

(b) Sulcal depth map



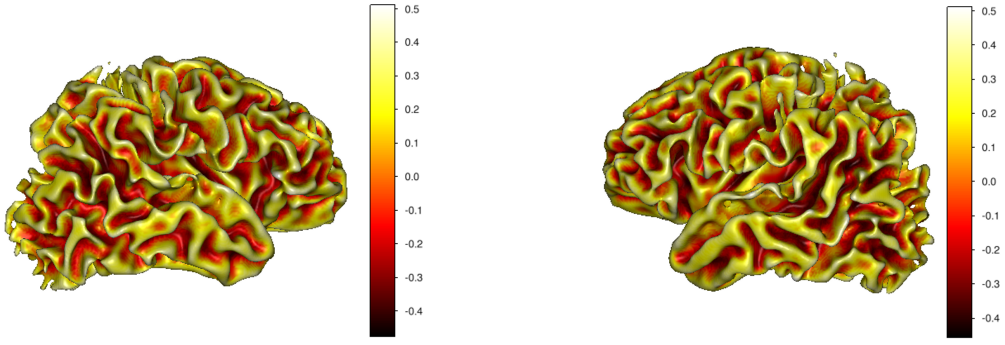
(c) Principal curvature: k_1

(d) Principal curvature: k_1



(e) Principal curvature: k_2

(f) Principal curvature: k_2



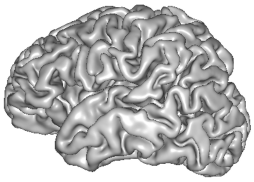
(g) Extrinsic Laplacian: $\sum_{i=1}^{3} \lambda_i$

(h) Extrinsic Laplacian

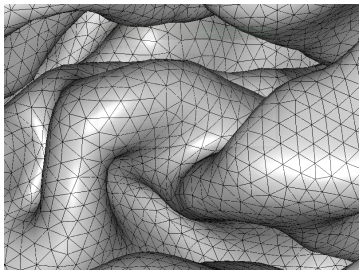
FIGURE 14: Obtained feature maps for the white matter of a dHCP subject. Left column (left hemisphere). Right column (right hemisphere). The curvatures are calculated for the white matter.

roradiology procedures," Medical image analysis, vol. 35, pp. 685–698, 2017.

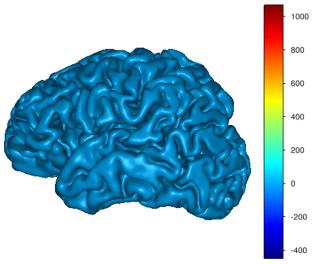
[15] F. Mémoli, G. Sapiro, and P. Thompson, "Implicit brain imaging," NeuroImage, vol. 23, pp. S179–S188, 2004.



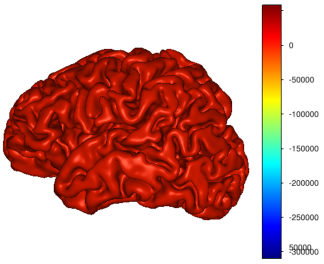
(a) Right hemisphere surface



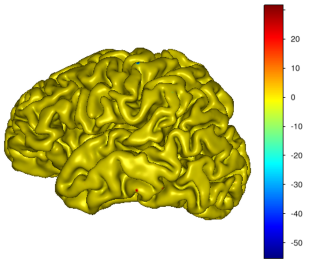
(b) Mesh quality



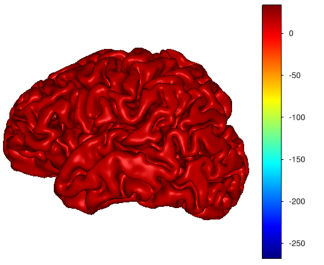
(c) Mean curvature (cubic order)



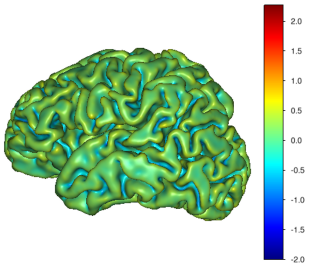
(d) Gaussian curvature (cubic order)



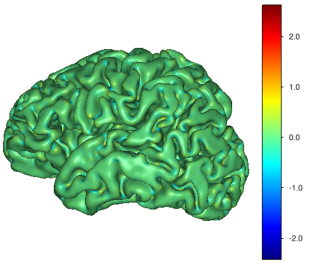
(e) Mean curvature (iterative fitting)



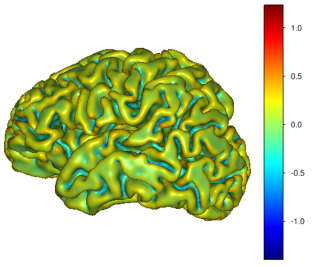
(f) Gaussian curvature (iterative fitting)



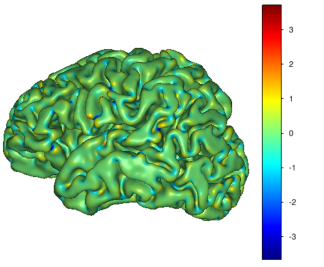
(g) Mean curvature (Rusinkiewicz)



(h) Gaussian curvature (Rusinkiewicz)



(i) Mean curvature (Trimesh)



(j) Gaussian curvature (Trimesh)

FIGURE 15: Curvature results for explicit methods: for the right hemisphere surface mesh generated by the dHCP structural pipeline.

[16] N. Paragios, "A level set approach for shape-driven segmentation and tracking of the left ventricle," IEEE transactions on medical imaging, vol. 22, no. 6, pp. 773–776, 2003.

[17] X. Lei, X. Yu, J. Chi, Y. Wang, J. Zhang, and C. Wu, "Brain tumor segmentation in MR images using a sparse constrained level set algorithm," Expert Systems With Applications, vol. 168, p. 114262, 2021.

[18] L. Á. Larios-Cárdenas and F. Gibou, "A deep learning approach for the computation of curvature in the Level-Set method," SIAM Journal on Scientific Computing, vol. 43, no. 3, pp. A1754–A1779, 2021.

[19] J. Dubois, M. Benders, C. Borradori-Tolsa, A. Cachia, F. Lazeyras, R. Ha-Vinh Leuchter, S. Sizonenko, S. Warfield, J. Mangin, and P. S. Hüppi, "Primary cortical folding in the human newborn: an early marker of later functional development," Brain, vol. 131, no. 8, pp. 2028–2041, 2008.

[20] L. R. Ment, D. Hirtz, and P. S. Hüppi, "Imaging biomarkers of outcome in the developing preterm brain," The Lancet Neurology, vol. 8, no. 11, pp. 1042–1055, 2009.

[21] R. Dahnke and C. Gaser, "Surface and shape analysis," in Brain morphometry. Springer Science+Business Media, LLC, 2018, pp. 51–73.

[22] J. E. Kline, V. S. P. Illapani, L. He, and N. A. Parikh, "Automated brain morphometric biomarkers from MRI at term predict motor development in very preterm infants," NeuroImage: Clinical, vol. 28, p. 102475, 2020.

[23] X. Wang, J. Lefèvre, A. Bohi, M. Al Harrach, M. Dinomais, and F. Rousseau, "The influence of biophysical parameters in a biomechanical model of cortical folding patterns," Scientific Reports, vol. 11, no. 1, pp. 1–14, 2021.

[24] O. Acosta, P. Bourgeat, M. A. Zuluaga, J. Fripp, O. Salvado, S. Ourselin, A. D. N. Initiative et al., "Automated voxel-based 3D cortical thickness measurement in a combined Lagrangian–Eulerian PDE approach using partial volume maps," Medical image analysis, vol. 13, no. 5, pp. 730–743, 2009.

[25] S. Budd, P. Patkee, A. Baburamani, M. Rutherford, E. C. Robinson, and B. Kainz, "Surface agnostic metrics for cortical volume segmentation and regression," in Machine Learning in Clinical Neuroimaging and Radiogenomics in Neuro-oncology. Springer Nature Switzerland, 2020, pp. 3–12.

[26] R. Santa Cruz, L. Lebrat, P. Bourgeat, C. Fookes, J. Fripp, and O. Salvado, "DeepCSR: A 3D deep learning approach for cortical surface reconstruction," in 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2021, pp. 806–815.

[27] R. V. Garimella and B. K. Swartz, "Curvature estimation for unstructured triangulations of surfaces," Los Alamos National Laboratory LA-03-8240, 2003.

[28] J. Goldfeather and V. Interrante, "A novel cubic-order algorithm for approximating principal direction vectors," ACM Transactions on Graphics (TOG), vol. 23, no. 1, pp. 45–63, 2004.

[29] D. Cohen-Steiner and J.-M. Morvan, "Restricted Delaunay triangulations and normal cycle," in Proceedings of the nineteenth annual symposium on Computational geometry, 2003, pp. 312–321.

[30] M. Kronenberger, O. Wirjadi, and H. Hagen, "Empirical comparison of curvature estimators on volume images and triangle meshes," IEEE transactions on visualization and computer graphics, vol. 25, no. 10, pp. 3032–3041, 2018.

[31] S. Osher and R. P. Fedkiw, Level set methods and dynamic implicit surfaces. Springer New York, 2005, vol. 1.

[32] A. J. Yezzi and J. L. Prince, "An Eulerian PDE approach for computing tissue thickness," IEEE transactions on medical imaging, vol. 22, no. 10, pp. 1332–1339, 2003.

[33] B. Rieger, F. J. Timmermans, L. J. Van Vliet, and P. W. Verbeek, "On curvature estimation of ISO surfaces in 3D gray-value images and the computation of shape descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1088–1094, 2004.

[34] L. Ronan, N. Voets, C. Rua, A. Alexander-Bloch, M. Hough, C. Mackay, T. J. Crow, A. James, J. N. Giedd, and P. C. Fletcher, "Differential tangential expansion as a mechanism for cortical gyrification," Cerebral Cortex, vol. 24, no. 8, pp. 2219–2228, 2014.

[35] A. Makropoulos, E. C. Robinson, A. Schuh, R. Wright, S. Fitzgibbon, J. Bozek, S. J. Counsell, J. Steinweg, K. Vecchiato, J. Passerat-Palmbach et al., "The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction," Neuroimage, vol. 173, pp. 88–112, 2018.

[36] M. Eyre, S. P. Fitzgibbon, J. Ciarrusta, L. Cordero-Grande, A. N. Price, T. Poppe, A. Schuh, E. Hughes, C. O'Keeffe, J. Brandon et al., "The Developing Human Connectome Project: typical and disrupted perinatal functional connectivity," Brain, vol. 144, no. 7, pp. 2199–2213, 2021.

[37] R. Sawhney and K. Crane, "Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains," ACM Transactions on Graphics, vol. 39, no. 4, 2020.

[38] J. A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge university press, 1999, vol. 3.

[39] N. Paragios, O. Faugeras, T. Chan, and C. Schnoerr, Variational, Geometric, and Level Set Methods in Computer Vision: Third International Workshop, VLSM 2005, Beijing, China, October 16, 2005, Proceedings. Springer-Verlag Berlin Heidelberg, 2005, vol. 3752.

[40] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," IEEE transactions on image processing, vol. 19, no. 12, pp. 3243–3254, 2010.

[41] E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," Journal of computational physics, vol. 210, no. 1, pp. 225–246, 2005.

[42] C. Avenel, E. Mémin, and P. Pérez, "Stochastic level set dynamics to track closed curves through image data," Journal of mathematical imaging and vision, vol. 49, no. 2, pp. 296–316, 2014.

[43] N. Sukumar and A. Srivastava, "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks," arXiv preprint arXiv:2104.08426, 2021.

[44] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," Journal of graphics tools, vol. 8, no. 2, pp. 1–15, 2003.

[45] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," Proceedings of the National Academy of Sciences, vol. 93, no. 4, pp. 1591–1595, 1996.

[46] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," Proceedings of the national academy of Sciences, vol. 95, no. 15, pp. 8431–8435, 1998.

[47] G. Peyré, M. Péchaud, R. Keriven, and L. D. Cohen, Geodesic methods in computer vision and graphics. Now publishers Inc, 2010.

[48] J. Hladuvka, A. Konig, and E. Groller, "Exploiting eigenvalues of the hessian matrix for volume decimation," Tech. Rep. TR-186-2-00-19, University of West Bohemia, 2001.

[49] A. Routier, N. Burgos, M. Díaz, M. Bacci, S. Bottani, O. El-Rifai, S. Fontanella, P. Gori, J. Guillon, A. Guyot, R. Hassanaly, T. Jacquemont, P. Lu, A. Marcoux, T. Moreau, J. Samper-González, M. Teichmann, E. Thibeau-Sutre, G. Vaillant, J. Wen, A. Wild, M.-O. Habert, S. Durrleman, and O. Colliot, "Clinica: an open source software platform for reproducible clinical neuroscience studies," in Annual meeting of the Organization for Human Brain Mapping-OHBM 2018, (Singapore, Singapore), 2018.

[50] E. Gibson, W. Li, C. Sudre, L. Fidon, D. I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu et al., "NiftyNet: a deep-learning platform for medical imaging," Computer methods and programs in biomedicine, vol. 158, pp. 113–122, 2018.

[51] C. Tor Diez, N. Passat, I. Bloch, S. Faisan, N. Bednarek, and F. Rousseau, "An iterative multi-atlas patch-based approach for cortex segmentation from neonatal MRI," Computerized Medical Imaging and Graphics, vol. 70, pp. 73–82, 2018.

[52] T. D. Gatzke and C. M. Grimm, "Estimating curvature on triangular meshes," International journal of shape modeling, vol. 12, no. 01, pp. 1–28, 2006.

[53] C. Engwer and A. Nüßing, "Geometric reconstruction of implicitly defined surfaces and domains with topological guarantees," ACM Transactions on Mathematical Software (TOMS), vol. 44, no. 2, pp. 1–20, 2017.

[54] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in Visualization and mathematics III. Springer-Verlag Berlin Heidelberg, 2003, pp. 35–57.

[55] B. Fischl, "FreeSurfer," Neuroimage, vol. 62, no. 2, pp. 774–781, 2012.

[56] E. M. Brown, M. E. Pierce, D. C. Clark, B. R. Fischl, J. E. Iglesias, W. P. Milberg, R. E. McGlinchey, and D. H. Salat, "Test-retest reliability of FreeSurfer automated hippocampal subfield segmentation within and across scanners," Neuroimage, vol. 210, p. 116563, 2020.

[57] M. J. Paldino, F. Golriz, W. Zhang, and Z. D. Chu, "Normalization enhances brain network features that predict individual intelligence in children with epilepsy," PloS one, vol. 14, no. 3, p. e0212901, 2019.

[58] N. Lehmann and U. Reif, "Notes on the curvature tensor," Graphical models, vol. 74, no. 6, pp. 321–325, 2012.

[59] V. S. Natu, M. J. Arcaro, M. A. Barnett, J. Gomez, M. Livingstone, K. Grill-Spector, and K. S. Weiner, "Sulcal depth in the medial ventral temporal cortex predicts the location of a place-selective region in macaques, children, and adults," Cerebral Cortex, vol. 31, no. 1, pp. 48–61, 2021.

[60] H. J. Yun, K. Im, J.-J. Yang, U. Yoon, and J.-M. Lee, "Automated sulcal depth measurement on cortical surface reflecting geometrical properties of sulci," PloS one, vol. 8, no. 2, p. e55977, 2013.

ACKNOWLEDGEMENT

The authors wish to express their thanks to Guillaume Auzias from Institut de Neurosciences de la Timone for providing the adult brain MRI data.