









## Comparative Study of 3D Reconstruction Methods for Medical Imaging

Loubna Lechelek<sup>1,3</sup> , Sylvain Gerbaud<sup>1,3</sup> , Rita Zrour<sup>1,3</sup> , Mathieu Naudin<sup>2,3</sup> ,  
Carole Guillevin<sup>2,3</sup>  and Sebastien Horna<sup>1,3</sup> 

<sup>1</sup> XLIM Laboratory, UMR CNRS 7252, University of Poitiers, France, [author@univ-poitiers.fr](mailto:author@univ-poitiers.fr)

<sup>2</sup> LMA Laboratory, UMR CNRS 7348, University of Poitiers France, [author2@chu-poitiers.fr](mailto:author2@chu-poitiers.fr)

<sup>3</sup> I3M, Common Laboratory CNRS-Siemens, University and Hospital of Poitiers, France,

Corresponding author: L. Lechelek, [loubna.lechelek@univ-poitiers.fr](mailto:loubna.lechelek@univ-poitiers.fr)

**Abstract.** The work presented in this paper is about comparison and analysis of four different reconstruction methods applicable to MR images. Two of them are based on the marching cube algorithm and the other two are contour based algorithms. We describe the studied reconstruction methods and compare their results. The comparative study examined, allows to establish similarity, equivalence, or distinctness between the four methods. We show that the various reconstruction methods produce different 3D models and each one has its own advantages and limitations.

**Keywords:** 3D Reconstruction, MR Imaging, Contour Based Reconstruction, Marching Cube.

**DOI:** <https://doi.org/10.14733/cadaps.2022.1000-1014>

## 1 INTRODUCTION

This paper focuses on 3D surface reconstruction techniques applicable to MR images. This area plays an important role in providing realistic 3D models of anatomy or structure of human being bodies, and is very useful for medical diagnosis, visualization, and model-based therapy planning. It can also be applied for other purposes like virtual education, surgical planning and further analyses. Imaging technology such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), typically provide 2D cross sectional images that are spaced evenly on individual slices. This data allows the reconstruction of a 3D model that can be used for many applications: visualization, diagnostic aid and simulation of the propagation of physiological elements. A precise 3D reconstruction model is important since it can reflect properly the structural characteristics of organs and ensures an accurate anatomical evaluation and precise surgical planning enabling thus higher efficiency in preoperative assessment. 3D surface reconstruction techniques extract polygonal surface from volume data which is obtained from cross-sectional MR images [23]. The primary goal to meet the clinical needs is to create a triangular mesh, which fits to the segmentation boundary inside the volume.

Historically, reconstruction techniques have been introduced with the famous Marching Cubes (MC) isosurface extraction algorithm [16]. The simplicity, robustness and rapidity of the marching cubes have made it the most widely used algorithm for interactive visualization of medical data. Recently, Schroeder *et al.* [27] introduced a fast isosurface extraction algorithm called Flying edges, which is the parallel Marching cubes algorithm implemented on multi-core CPUs. However, the extracted surface may present discontinuities and topological incoherences due to the ambiguities in the interpolant behavior [7, 21]. This is the reason why several authors have increased the number of entries in the marching cubes triangulation look up table [4, 6, 15, 19, 20] to resolve the ambiguities and thus enhance the topological correctness [5]. Marching cube algorithms might generate also meshes which are not homeomorphic and seems inconsistent across cell borders, in order to resolve the latest problem an asymptotic decider has been introduced by Nielson and Hamann [21] to resolve ambiguities at the cells faces. An alternative strategy to generate 3D closed surface model is to reconstruct quad only meshes from volume data which is the purpose of the dual marching cube algorithm (DMC) introduced in [12].

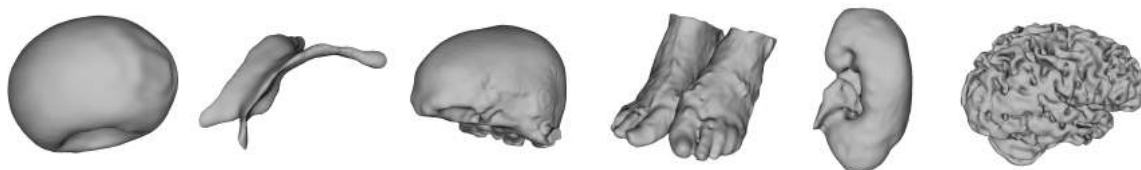
Besides Marching cubes, numerous contour-based surface reconstruction techniques have been developed, and can be classified into two main classes: contour stitching that is based on explicit surface representation and volumetric methods that employ implicit representations [3]. The former interpolate between the adjacent cross-sectional contours using a mesh [1, 2, 8, 9, 11, 18]. The disadvantages of these methods is that they use a mesh composed only of straight lines and flat faces, which does not allow to depict efficiently smooth biological data. Furthermore, these early techniques do not deal with handling special cases such as keyholes contours, rapid changes and branching. These issues have been addressed recently by Sunderland *et al.* [29] in a more robust algorithm for interpolating planar contours. The second class of methods can be further divided into two main categories: *interpolating methods* in which the cross-sectional contours are expressed by smooth and continuous 2D distance fields [10, 13, 14, 17, 25, 31] and *approximating methods* that tend to fit an implicit function to the input data using provided normal such as Multi-level Partition of Unity (MPU) implicit models proposed by Braude *et al.* [3] inspired from the work of Ohtake *et al.* [22].

In this paper, we present a comparison between four different reconstruction methods, two of them are based on the marching cube algorithm [12, 27] and the other two are contour based algorithms [3, 29]. The comparison allows to establish similarity, equivalence, or distinctness between the four methods. The main purpose is to measure the accuracy of each method, determine if the 3D reconstructed model is precise, faithful to the data provided by the segmentation and meet the medical expectations. If these factors are met the model can be used to facilitate or optimize diagnosis, surgical planning and virtual biopsy. In the long term, having an accurate model can offer a non-invasive technique to detect and localize the presence of a tumor. We evaluate the quality of each of the reconstruction method. All our tests and comparison are in 3D Slicer [24] and Slicer RT (extension that has been developed for the 3D slicer platform which provides among other research workflows, 3D surface reconstruction from a set of contours) [26]. For this study, we have first reconstructed the models shown in Fig. 1 from the sample datasets included with 3D Slicer. We also applied the four methods on human cerebral cortices derived from real MRI clinical dataset.

This paper is organized as follows. In section 2 we present the four studied reconstruction methods. In section 3 we expose the comparison and analysis between the four methods. Section 4 details a discussion, advantages and limitations between the four methods. Finally section 5 states some conclusion and perspectives.

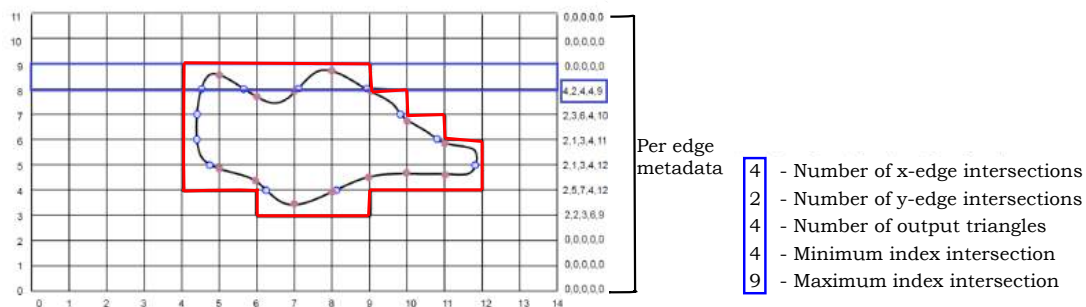
## 2 3D RECONSTRUCTION METHODS

The problem of reconstructing 3D models has been studied since the 1970's [8, 11, 16]. Numerous methods and techniques have been proposed since that time, and fall into two main classes: marching cube algorithms and contour based algorithms.



**Figure 1:** 3D reconstructed models from MRI images used for the study.

This section describes briefly the four reconstruction methods studied in this work : Flying Edges (FE), Dual Marching cube (DMC), SlicerRT contour interpolation (CI) and MPU implicit models (MPU). Two of the methods are based on marching cube algorithm (FE and DMC) and the other two are contour based algorithms (CI, MPU). All the four methods are tested, visualized and analyzed using an open-source software package 3D Slicer [24] and Slicer RT (source extension that has been developed for the 3D Slicer platform). 3D Slicer provides among others, tools for reconstructing and visualizing triangulated surface models. In the following, a description of each of the four methods is recalled.

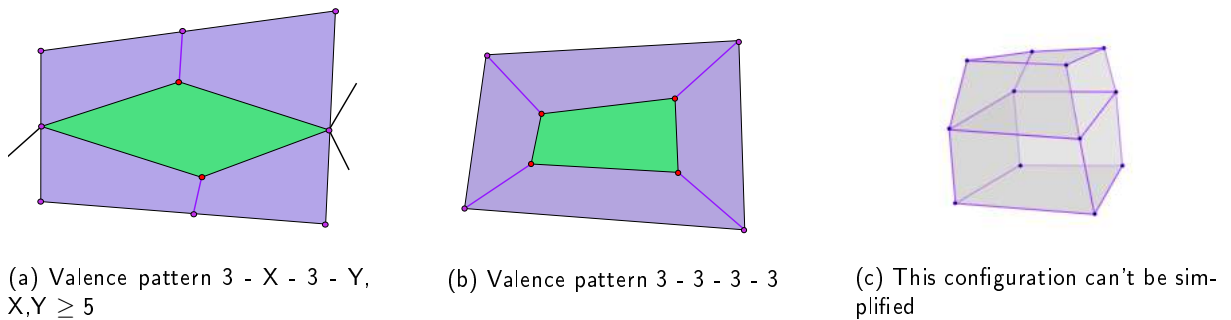


**Figure 2:** The region of the grid processed at the end of the second stage; On the right side is a metadata array describing the interaction of the isosurface with the grid edges.

## 2.1 Flying Edges (FE)

The Flying Edges (FE) is an algorithm designed from the standard MC algorithm, in a way that is better optimized for multi-core processors [27]. It operates on a three dimensional grid of voxels and uses a lookup table from which the triangle configuration of each voxel is determined. The location and gradients of the surface vertices are approximated in the same way as MC. The difference between the two algorithms lies in how the volume data is processed. Whereas, the MC marches through all the voxels of the volume in one pass, FE uses multiple preprocessing passes. The first preprocessing stage marches through all the voxels along one dimension of the volume, determines which edges are intersected by accessing the isovalues stored in the volume and counts the number of intersections. The sections are crossed in parallel and each indicates its first and last intersections, so that only a smaller part of the grid will be processed in the next stages. The second preprocessing stage determines which edges along the other two dimensions are intersected. However, instead of processing the entire grid and accessing the scalar values of the volume, the second preprocessing step uses the informations gained in the first preprocessing stage to determine which of the remaining edges are intersected. For example as shown in the Fig. 2, after the second pass, the grid (in 2D) is restricted to a small region circumnavigating the isosurface. By directly exploiting informations in the first preprocessing

stage, the computation time is being reduced and the cache performance is being maximized. The third of the preprocessing stages allocates memory to contain the triangles and points, while the last stage calculates the intersection points and generates triangles. 3D slicer uses VTK's implementation for FE and adds an additional mesh smoothing post-processing step [30] to remove staircase artifacts and to exhibit the natural look of anatomical structures. We have used this process to generate models shown in Fig. 1.



**Figure 3:** DMC : quad and valence pattern (a) valence pattern 3 - X - 3 - Y where  $X, Y \geq 5$  vertices in red are collapsed into a new vertex, removing the green element; (b) valence pattern 3 - 3 - 3 - 3; in the middle, the edges in green are collapsed moving the vertices in red toward the vertices in purple and the green elements are removed; (c) for the configuration on the right no element will be removed in order to keep the mesh manifold.

## 2.2 Dual Marching Cubes (DMC)

The Dual Marching Cube (DMC) is an alternative strategy to generate 3D closed surface model by reconstructing quad only meshes from volume data. Quad faces are supposedly better shape than triangle ones. DMC relies on the lookup table introduced in [20] to produce a quad only mesh from a volume data such as a 3D grid. In our study, we used a novel algorithm using DMC [12] that consists in the reconstruction of that kind of mesh and a topological simplification to reduce the number of irregular vertices. This particular algorithm does not rely on a lookup table to create quads. The output of this algorithm is topologically consistent across cell borders, i.e it is watertight. The DMC produces a quad only mesh stored in a halfedge data structure. The execution flow of this algorithm is divided in three main steps : 1) initialize buffers (for parallelize purpose); 2) compute the DMC mesh; 3) generate a halfedge data structure.

The computation of the DMC quadrilateral mesh is computed with two CUDA kernels. Each thread started by the first kernel carries out three additional steps for the cell being processed.

The first one consists of the computation of up to four MC polygons. A segment is given by intersecting a face with the iso-surface, and these segments are connected to build up closed polygons. Based on the standard cases of the MC algorithm, it uses an asymptotic decider [21] to resolve ambiguous cases. The second step is the computation of a vertex, within the cell, that is a representative of the iso-surface computed for each surface branch. Initially it is placed as close as possible to the iso-surface, and then is moved iteratively towards the surface. Additional data are calculated, such as normals at the vertex representative. The last step is the computation of quadrilaterals by connecting vertex representative from four neighbor cells sharing a common edge. Quads are consistently oriented depending on the edge configuration used to build them. In contrast of the standard MC lookup table, quads are oriented such that their normals point in the same direction as the gradient of the volume data. An halfedge data structure is computed to store all quads data generated.

At the end, due to the transitions between layers in the volume data, the DMC mesh has numerous quads

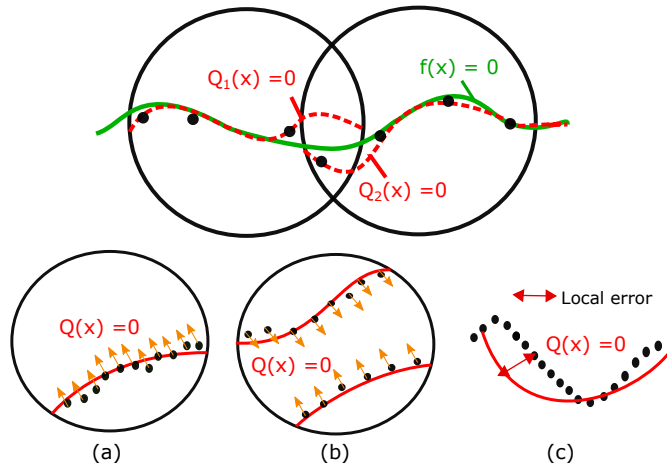
with particular valence pattern : 3 - X - 3 - Y, where  $X, Y \geq 5$ ; 3 - 3 - 3 - 3. This elements can be easily removed from the mesh to simplify it (see Fig. 3). The output can be smoothed by the same exact method as the FE inside 3D Slicer.

### 2.3 Contour Interpolation (CI)

The Contour Interpolation (CI) algorithm presented in [29] and implemented in SlicerRT, is a method used for creating 3D closed surface models from planar contours. The main idea of the algorithm is to interpolate between the adjacent cross-sectional contours using a mesh, going through four main stages. The first stage removes the keyholes from the set of contours to avoid errors in the triangulation process. Keyholes issues appear where the inner and outer contours are connected within a channel and considered as one contour. The algorithm handles the keyholes by comparing each point in the contour with the remaining points in the same contour and adds additional contours if two points in the contour have been found to have violated the threshold distance criterion. The second stage called correspondence determines which contours should be connected by triangles between each of the slices. The algorithm considers each pair of contours as a connected structure during the triangulation process if the corresponding bounding boxes overlap. The third stage called branching computes the branching patterns for contours that possess a correspondence to multiple contours on an adjacent slice. In that case, the larger contour is divided into separate sections and each section is marked to be triangulated with the closest small contour located on the adjacent layer. The last stage called tiling joins between the neighbouring cross-sectional contours by creating a strip of triangles using the correspondences between the contours vertices. It is based on the variant PointWalk algorithm that triangulates the mesh by minimizing the length of the edges spanning the contours with a dynamic programming method [28]. Once the triangulation is finished, the algorithm seals the external contours to fill the gaps that are located on the top and the bottom of the 3D model using the Delaunay triangulation which constructs the final closed surface.

### 2.4 MPU Implicit Models

Presented in [3], the MPU interprets the contours as points in  $R^3$  and employs Multi-level Partition of Unity (MPU) implicit models to create a surface that approximately fits to the 3D points. The main idea of the algorithm is to subdivide the point set in space using a recursive octree-based subdivision scheme, where the surface estimation of each partition is performed locally and the overlapping local implicit functions are blended together to produce the overall surface (i.e zero iso-surface from which the mesh is then extracted). Figure 4 briefly recalls the principle. Since the MPU requires surface normal information, the surface normal at each contour point is estimated from the binary volume, constructed from the set of contours as described in [3]. The MPU implicit functions are approximated using local piecewise quadric functions, where the choice of the appropriate function depends on local surface features implied by the point normals. They can either be a 3D quadric, a bivariate quadric polynomial, or a piecewise quadric surface. The choice of one of the three functions is governed by examination of the surface normals associated with the points in each partition. If all normals are oriented in the same direction then a bivariate quadric is used (Fig. 4a), otherwise a general 3D quadric is used (Fig. 4b) while the piecewise quadric is generally employed to detect edges and corners. At each stage of the subdivision process, the local function's accuracy has to be evaluated and compared to a user-specified tolerance value ( $tol$ ) in order to determine if the local implicit surface is needed to be refined (Fig. 4c) and thus allows to refine areas of higher detail. The tolerance parameter controls the quality and smoothness of the reconstruction and ensures that the reconstructed surface lies within ( $tol$ ) distance of the input point data. This method is implemented and added as a loadable module in SlicerRT.



**Figure 4:** Principle of MPU implicit Models. Top: Two local approximations (dashed red curves) blended to form the global MPU function (solid green curve). Bottom: (a-b) quadric functions are used to approximate the local shape; (a) bivariate quadric is used; (b) general 3D quadric is used; (c) the local function's error is evaluated in each step of the algorithm in order to determine if the local function has to be refined.

### 3 COMPARISON AND ANALYSIS

This section describes the models used for this work and details the comparison and analysis between the four reconstruction methods showing thus advantages and limitations for each.

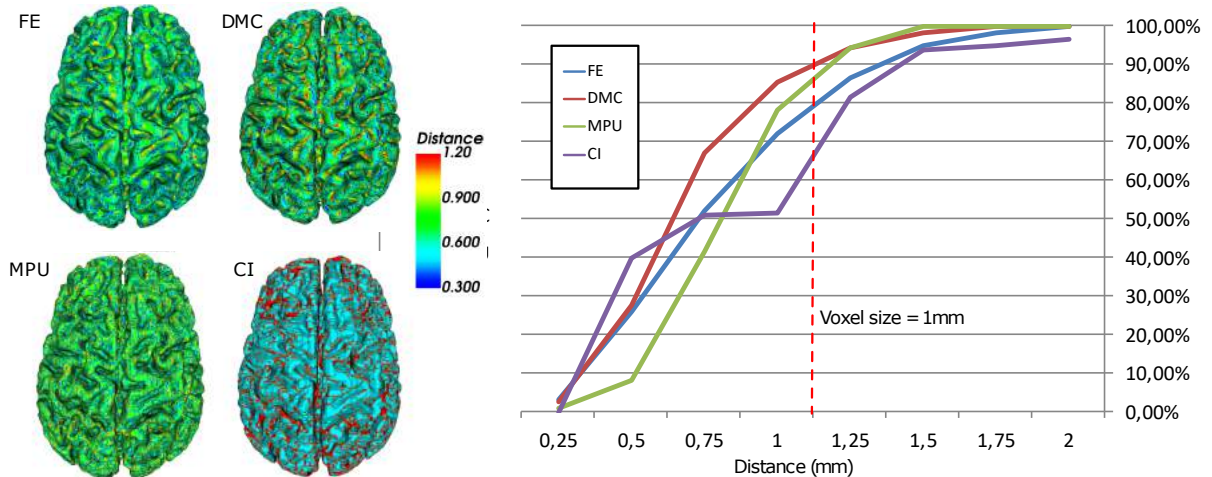
#### 3.1 Strategy And Models

In order to compare the four different methods in the same software, we have implemented the MPU algorithm for creating 3D closed surface models from planar contours in SlicerRT and added also the DMC in 3D Slicer. The aim of the reconstruction methods studied is to produce a 3D model as close as possible to the data provided by the segmentation process. For each 3D model, we evaluate the distance between the segmentation and the produced mesh (i.e. the distance between each point of the segmentation to the nearest point of the 3D model). To compute the distance, we have added a comparison plugin to 3D slicer based on Hausdorff distance. Distance computation allows to evaluate the quality of reconstructions and determine how faithfully each model fits to the segmentation. We have also added a quantitative study based on curvature analysis of the reconstructed models. The curvature behaviour conveys information about local surface shape and provides some insight about the amount of smoothness.

We have tested and evaluated the reconstruction methods with a number of datasets (Brain, Skull-Bone, ventricle and Kidney). The first three ones are 3D slicer sample datasets while the Kidney dataset is provided by the CHU of Poitiers. For these samples, the resolution in the  $X$  and  $Y$  direction is different from that in  $Z$  direction. We have also compared and evaluated the methods on forty human cerebral cortices dataset also provided by our clinical partner. For these samples the resolution is the same in all the three directions.

#### 3.2 Comparison And Measures

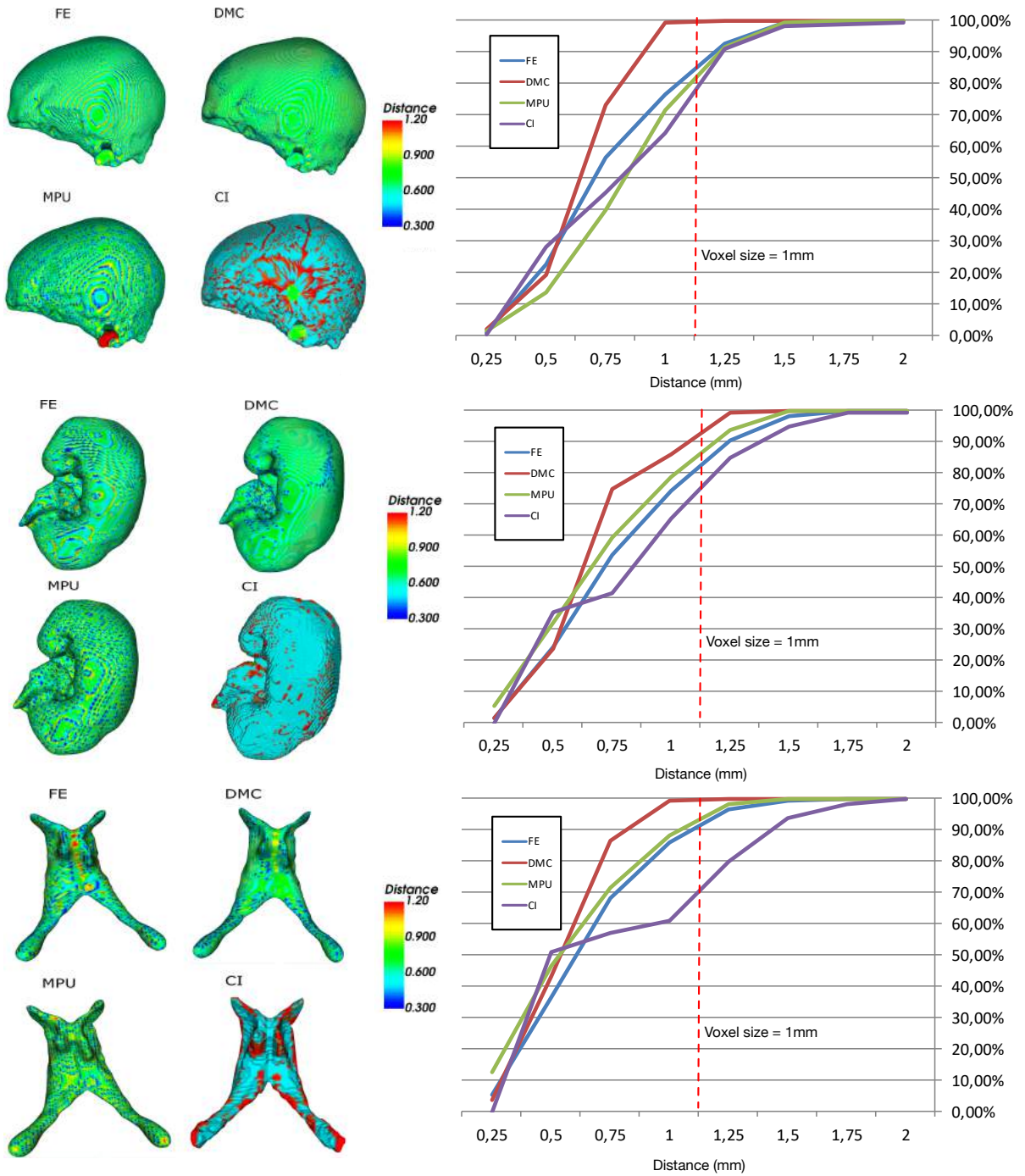
Fig. 5 and Fig. 6 present the results of the distance measure on the brain, skull-Bone, Kidney and ventricle models for each of the four reconstruction methods. For each model, by examining each of the corresponding four reconstructed models, one can notice a predominant green color (distance around the voxel size of 1mm)



**Figure 5:** Visualization of the distance between segmentation and 3D model. The voxel size of the segmentation grid in XY direction is of 1mm: (left) distance computed with hausdorff metric (blue reflect low distances and red high distances). (right) cumulative percentage of the proportion of element as a function of their distance to the segmentation.

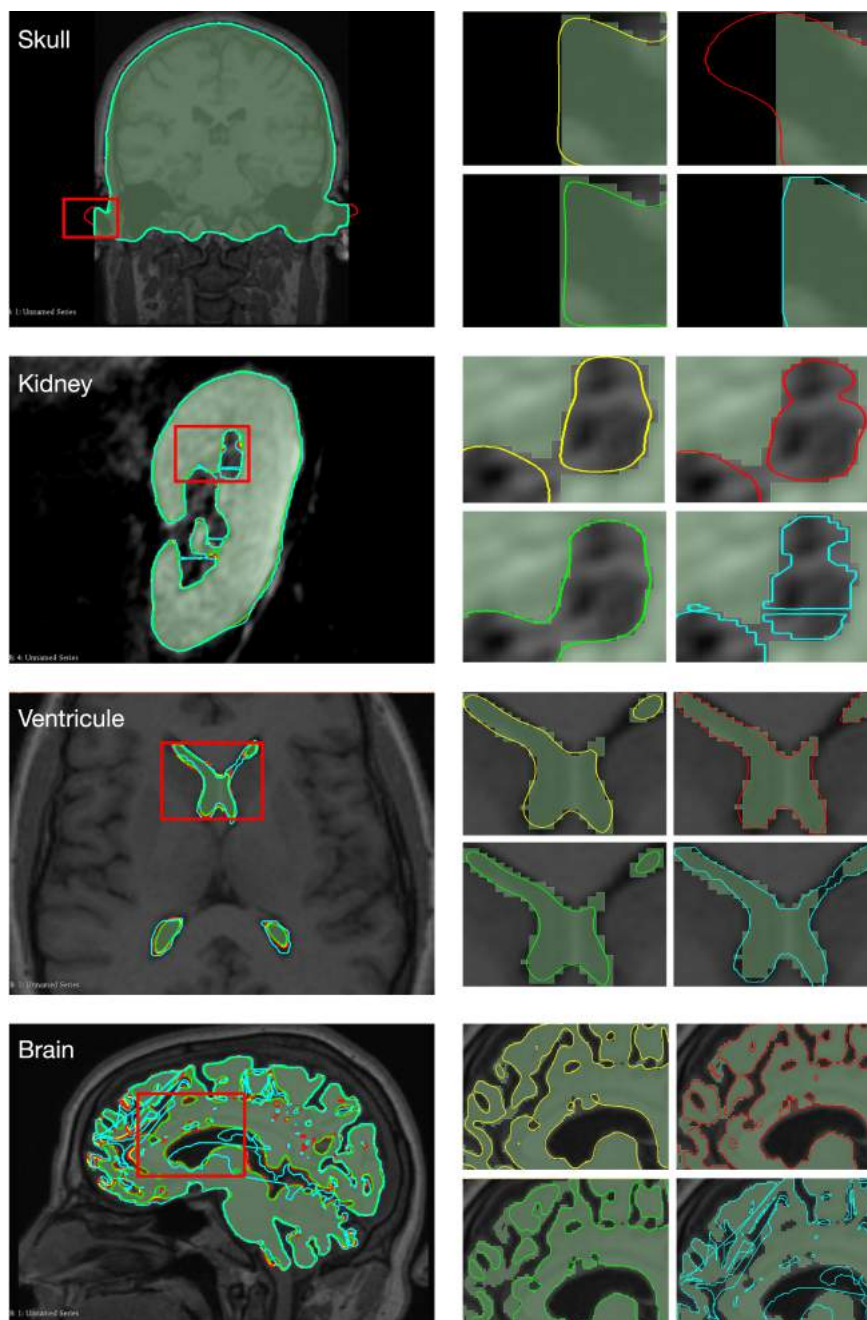
for the FE, DMC and MPU methods. One can notice also that the blue (very small distances) and red (very large distances) areas are much less dominant and not positioned at all at the same place when moving from one model to another. As for CI the results are quite different since a predominance of a cyan color (distance less than half voxel size) and red (greater than voxel size) is observed. Results can be deepened while looking to the curves that show the distance versus the cumulated percentage of points. Concerning FE, DMC and MPU, the majority of the surface points lies within one voxel of the segmentation (Brain: FE-72.08%, DMC-85.52%, MPU-78.48%), (Skull-Bone: FE-76.59%, DMC-99.51%, MPU-71.57%), (Kidney: FE-74.09%, DMC-85.78%, MPU-79.03%) and (Ventricle: FE-85.85%, DMC-99.64%, MPU-88.49%). As for CI, only 51.61% of its points for the Brain are around the voxel size. The Skull-Bone, Kidney and Ventricle datasets perform even better with 64,28%, 61,02% and 65,17% respectively of points around the voxel size. However, if we consider smaller distances less than half a voxel size, CI has the greater percentage of points for all models (Brain: CI- 40%, 30% for DMC and FE and less than 10% for MPU.), (Skull-Bone: CI- 28%, DMC-23% and less than 20% for FE and MPU), (Kidney: CI-35%, MPU-32% and less than 25% for DMC and FE) and (Ventricle: CI-51%, MPU-46%, FE-36%, DMC-43%).

Fig. 7 shows a cross section of the four reconstruction methods, in the left part all the four models are superimposed with different colors to see them together. On the right part, a zoom on a specific zone (red square) is performed in order to examine the reconstruction details. One can notice that the reconstruction results are different depending on the regions. For the skull-Bone, we see that MPU extrapolates and step aside the segmented region, The DMC is more closer than MPU but less precise than FE and CI. As for the brain, due to the complexity of the structure the CI is not precise and generates messy reconstruction and wrongly reconstructed parts. FE and MPU appear more precise than DMC for this specific region. Looking into the kidney in Fig. 7, one can remark that CI constructs three components while FE and MPU two components and DMC found one component on this zoomed region. Lastly, for the ventricle, one can remark that MPU is more precise for this specific region than FE and DMC and CI connect some distant regions.



**Figure 6:** Visualization of the reconstruction errors of each of the four reconstruction methods : FE, MPU, DMC and CI applied on the Skull-Bone, Kidney, and Ventricle; The voxel size of the segmentation grid in XY direction is of 1mm; (left) reconstruction error from blue (low) to red (high) using Hausdorff metric. (right) cumulative percentage of the proportion of element as a function of their distance to the segmentation..



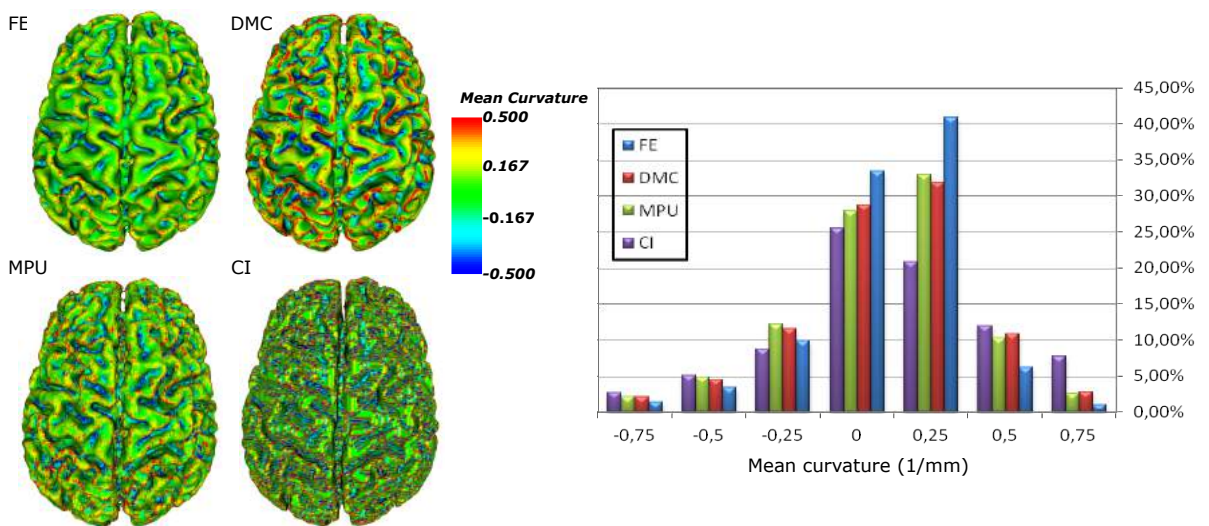


**Figure 7:** Reconstruction of the Skull-Bone, Kidney, Ventricle and brain with FE (yellow), MPU (red), DMC (green) and CI (cyan); (left) overlaid models; (right) a zoom on the red rectangle of left showing the reconstruction result of each method.

Sub	FE				DMC				MPU				CI			
	min	max	St dev	% < 1.0	min	max	St dev	% < 1.0	min	max	St dev	% < 1.0	min	max	St dev	% < 1.0
1	0.0058	2.5115	0.35	78.41	0.0079	2.3001	0.26	90.81	0.0039	7.3699	0.36	78.73	0.0999	11.367	0.53	48.73
2	0.0104	2.4756	0.34	78.96	0.0108	2.1415	0.25	91.34	0.0119	4.6646	0.35	79.89	0.0997	5.1546	0.41	48.66
3	0.0061	2.4166	0.35	76.48	0.0097	2.1581	0.26	89.72	0.0054	4.8169	0.34	81.12	0.0996	4.9284	0.42	48.12
4	0.0025	2.4276	0.34	79.32	0.0111	2.1271	0.25	91.69	0.0091	5.0137	0.35	79.10	0.0998	5.6613	0.42	49.86
5	0.007	2.3833	0.34	79.31	0.016	2.5651	0.30	87.71	0.0925	5.8206	0.35	81.35	0.0999	30.364	3.81	44.19
6	0.0092	2.4204	0.34	78.52	0.0107	2.2041	0.25	91.24	0.0027	4.9697	0.34	81.33	0.0998	5.0249	0.41	48.56
7	0.0066	2.3533	0.34	79.13	0.0146	2.1484	0.24	91.85	0.0024	5.4966	0.34	80.68	0.0999	7.2034	0.41	49.53
8	0.0067	2.3533	0.34	79.13	0.0146	2.1484	0.24	91.85	0.0025	4.6214	0.34	80.69	0.0999	34.480	3.96	20.63
9	0.011	2.5563	0.35	78.35	0.0108	2.6516	0.30	86.94	0.0102	4.2566	0.34	80.59	0.0997	5.1156	0.40	49.63
10	0.0038	2.3842	0.36	76.29	0.0047	2.5403	0.32	85.09	0.0124	4.3403	0.34	82.89	0.0998	10.2709	0.47	48.01
11	0.0025	2.3755	0.34	79.88	0.0094	2.1311	0.24	92.22	0.0099	5.3919	0.34	79.64	0.0996	12.283	0.61	49.73
12	0.004	2.5685	0.35	76.51	0.006	2.349	0.26	90.18	0.0116	4.1806	0.35	80.73	0.0999	4.8316	0.42	47.88
13	0.0093	2.5321	0.35	76.56	0.0088	2.1998	0.26	90.20	0.0049	5.0937	0.35	80.33	0.0999	7.6583	0.51	42.96
14	0.0119	2.4490	0.34	78.32	0.0105	2.1483	0.26	90.32	0.0104	4.9786	0.33	82.49	0.0999	15.922	0.93	46.77
15	0.0152	2.3795	0.34	78.57	0.0116	2.1447	0.25	91.45	0.008	7.2081	0.34	82.10	0.0999	7.3545	0.46	44.25
16	0.0156	2.4530	0.35	78.32	0.0137	2.1726	0.25	91.13	0.0095	5.2461	0.34	82.14	0.0995	9.9111	0.47	48.48
17	0.0160	2.4620	0.35	78.39	0.0147	2.2408	0.25	91.20	0.0106	5.1666	0.33	82.76	0.0996	50.545	10.63	34.13
18	0.0034	2.3358	0.34	78.67	0.0125	2.1753	0.25	91.24	0.0173	5.0233	0.35	78.81	0.0997	7.0357	0.40	50.30
19	0.0029	2.5603	0.35	78.94	0.0104	2.1483	0.26	90.96	0.0032	4.2001	0.35	80.39	0.0997	36.554	5.92	40.37
20	0.0073	2.4643	0.36	75.97	0.0081	2.1585	0.27	89.16	0.0104	7.2795	0.36	79.68	0.0999	4.9689	0.42	48.23
21	0.0096	2.4802	0.35	76.86	0.0137	2.1462	0.26	90.16	0.0108	4.3826	0.35	79.87	0.0999	21.4684	1.63	45.19
22	0.0052	2.4678	0.36	76.51	0.0132	2.1849	0.26	89.67	0.0144	4.4069	0.34	81.13	0.0999	4.6914	0.42	47.57
23	0.0032	2.5602	0.34	80.12	0.0080	2.1434	0.24	92.24	0.0117	4.6251	0.33	82.75	0.0999	65.469	15.93	28.68
24	0.0050	2.3940	0.35	78.87	0.0083	2.2076	0.25	91.24	0.0111	4.2750	0.35	80.39	0.0999	25.5963	2.59	45.41
25	0.0155	2.4205	0.36	75.65	0.0166	2.1805	0.27	89.28	0.0048	4.3083	0.33	83.03	0.0999	14.361	0.56	47.84
26	0.0067	2.4463	0.35	80.32	0.0071	2.2116	0.26	91.16	0.0149	6.9129	0.37	76.31	0.0997	7.4195	0.42	49.51
27	0.0151	2.5322	0.34	78.73	0.0122	2.1941	0.25	91.41	0.0140	4.8852	0.35	78.00	0.09987	22.8671	1.19	49.51
28	0.0075	2.4959	0.34	80.57	0.0102	2.1884	0.24	92.08	0.0052	4.9919	0.33	83.12	0.0996	29.901	3.38	45.18
29	0.0157	2.5447	0.34	79.92	0.0075	2.2450	0.24	92.21	0.0120	3.8786	0.33	80.80	0.0998	6.9188	0.44	49.02
30	0.0086	2.4741	0.37	76.42	0.0144	2.1592	0.28	88.11	0.0093	5.2818	0.36	78.76	0.0998	18.535	1.26	47.43
31	0.0072	2.4841	0.34	79.36	0.0172	2.1080	0.25	91.31	0.0032	4.1038	0.35	79.62	0.0996	20.416	1.37	47.66
32	0.0098	2.4173	0.35	77.78	0.0013	2.1695	0.26	90.57	0.0055	5.4231	0.35	79.65	0.0997	4.8466	0.41	48.89
33	0.0125	2.3255	0.35	78.42	0.0102	2.1039	0.26	90.91	0.0110	4.3413	0.35	80.04	0.0999	17.4163	0.81	49.60
34	0.0128	2.5866	0.34	79.50	0.0054	2.8057	0.34	79.50	0.0032	4.8449	0.35	79.75	0.0999	5.2354	0.40	50.54
35	0.0025	2.432	0.35	78.03	0.0104	2.1746	0.26	90.45	0.0073	4.1964	0.36	78.38	0.0996	15.260	0.95	48.31
36	0.0100	2.4668	0.34	79.65	0.0093	2.1473	0.24	92.05	0.0020	4.2403	0.35	79.25	0.0997	27.4243	2.77	45.90
37	0.0105	2.4886	0.34	79.68	0.0065	2.2411	0.24	92.10	0.0095	4.2359	0.33	81.35	0.0998	23.271	1.84	46.43
38	0.0120	2.4996	0.34	79.65	0.0123	2.2135	0.24	92.01	0.0066	4.0263	0.34	80.84	0.0998	39.711	5.83	41.83
39	0.0099	2.5446	0.35	77.09	0.0112	2.2276	0.26	90.26	0.0049	8.4550	0.37	81.24	0.0996	4.7297	0.40	48.65

**Table 1:** Approximation quality of reconstructed surfaces for 39 brain cortices. Metrics are calculated in units of voxels (1mm).

In order to enlarge, validate, and evaluate the effectiveness of the reconstruction methods, we tested each of them on a large cerebral cortices with intricate geometries. The statistics related to the complete reconstruction are calculated. The minimum, maximum, and standard deviation of the distance values for each reconstruction method are given in Table 1, as well as the percentage of points that are one voxel size away from the segmentation. All of the metrics that are presented here are in voxel units (1mm). For example, a maximum distance of 2 signifies that the point surface lies at most 2 voxels away from the segmentation. Concerning DMC and FE, despite the maximum distance values lie between 2 and 3 voxels, the percentage of points indicates that almost all regions of the reconstructed surfaces lie within the voxels defined by the segmentation. On average, 90.33% of the surface points for the DMC and 78.39% of the surface points for the FE are around the voxel size of 1mm. As for MPU, while the maximum distance value can reach 8 voxels for some cortices, on average, 80.51% of its points are less than a voxel size (1mm) away from the segmentation. This demonstrates that DMC, FE and MPU offer a sub-voxel accuracy to the data provided by the segmentation. The results are quite different for CI, since the maximum distance values are important for the majority of cortices. Only 45.95% of the surface points on average for CI are around the voxel size. Results can be further deepen looking at the standard deviations where the statistics show that the standard deviations of the DMC, FE and MPU still stay relatively low compared to CI (On average: DMC- 0.26, FE- 0.35, MPU- 0.35, CI- 1.91). Besides that, looking at minimum distances, one can notice that the minimum distance values of CI are around 0.1 for all cortices. This means that the reconstruction of CI always start at 0.1 mm away from segmentation, unlike the other methods where the minimum distances are smaller.



**Figure 8:** Visualization of the mean curvature on a 3D brain model: (left) mean curvature estimated by VTK (triangles are color-coded by curvature; blue reflects low curvatures and red high curvatures). (right) percentage of points as a function of their mean curvature value.

Besides the qualitative study, we have also quantitatively evaluated and compared the reconstruction methods through analysing the curvature information of the reconstructed surfaces. As shown in figure 8, for each 3D brain model, we have estimated the curvature at each point of the produced mesh using the `vtkCurvatures` class from the VTK Library [28]. Fig. 8 presents the results of the mean curvature measure on each point of the brain model for each of the four reconstruction methods. The results show that the surface curvature varies strongly from one model to another. On the left side of the figure, one can observe that the green color is predominant for the four methods while the red color (very high curvature) is much less

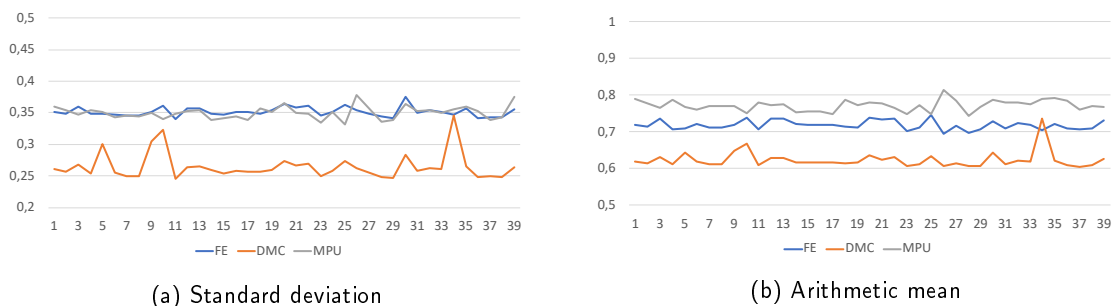
dominant for the FE compared to DMC, MPU and CI. This means that in overall the curvature surface of the FE is relatively low compared to DMC, MPU or CI. Results can be validated looking at the statistics on the right side of the Fig. 8 that show the mean curvature versus the percentage of points. If we consider low curvatures (between -0.25 and +0.25) the FE has the greater percentage of points; however this discrepancy vanished with higher curvatures where the FE has the smaller percentage of points, followed by MPU, DMC then CI. This is mainly due to the fact that FE produces no smooth surfaces, so as the surface triangles tend to follow the voxel boundaries resulting in a step-like surface. The mesh smoothing post processing applied on the output model gives a smooth effect but in return it tends to change the local surface shape and smooth greatly areas with high curvatures. Unlike the FE, DMC produces slightly smooth surfaces thanks to its algorithm which plays on the valence of the faces and employs a topological simplification to reduce the irregular vertices and thus it does not require a strong smoothing which may impact the local surface shape. As for the MPU, since it is based on adaptive implicit functions which confine the reconstruction to a specified user tolerance parameter, the amount of smoothing applied to the data is still controllable and thus the local surface shape depends on the choice of the tolerance value. The CI process produces more high curvatures (greater than +0.5), and less flat curvatures. This is due to the fact that the model has not undergone mesh post-processing smoothing, so areas with high curvatures are maintained.

#### 4 DISCUSSION

The results exposed in the previous section showed that the reconstruction methods behave differently and have distinct reconstruction results depending on the regions.

MPU appeared to be very precise for some regions but extrapolated some others increasing thus the maximum error as shown in Table 1 where its maximum error appear to be greater than FE and DMC but less than CI. MPU has the advantages to be more precise, since it is based on adaptive implicit functions which conform to local features and details. Another property that can be ensured by the MPU is the continuity of the reconstruction since it provides smooth and continuous surfaces with free staircases artifacts without the need for a mesh post-processing smoothing. One of the major drawback of the MPU is the integration in the implicit functions of both the flexibility of specifying the smoothness of a region and the accuracy of the shape approximation to the data provided by the segmentation. To achieve a high level of smoothness, a relatively large value of the tolerance has to be used, which will subsequently result in a poor approximation of the original shape of the segmentation so the user must tune the tolerance parameter to produce visually plausible results with reasonable error statistics.

As for FE and DMC they appear to be close in terms of overall results. However one can notice that the FE has an overall smaller minimal distance than the DMC, while the DMC has greater max distance than the FE. Examining the standard deviation (Fig. 9a), one can remark that DMC has the smaller one, while the FE and MPU are very close. Looking into the arithmetic mean (Fig. 9b), the DMC has the lower one, followed by FE and then MPU. The major drawback of the FE is derived from the pre-configured look up table which give rise to a number of topological ambiguities and thus it may output surfaces which are not necessarily coherent with the geometric configuration of the segmentation. This problem is overcome by DMC which does not rely on a lookup table but uses an asymptotic decider to resolve this ambiguities. It uses a topological simplification to reduce the number of irregular vertices. Another difficulty of MC based methods is related to the mesh post processing smoothing, since MC based methods generally produce models with staircases and terraces artefacts. The smoothing treatment allows to exhibit the natural look of the anatomical structures but in return, it may degrade the accuracy of the reconstruction.



**Figure 9:** FE, DMC and MPU reconstruction computed on each of the 39 cerebral cortices of Table 1; CI is not represented owing to its high standard deviation.

The advantages of CI is that it gives the greater percentage of points having distances less than half a voxel size (Fig. 6). The problem of CI occurs essentially, in the case of segmentation with intricate geometry, where the contours are not similar in shape and size, as well as in the presence of multiple inner contours, which is generally the case of cerebral substances. These cases often incorrectly alter the correspondence between contours and thus triangulate between the inappropriate contours (see Fig 7). We should also mention that the standard deviation of CI is higher than the other three (see Table 1).

## 5 CONCLUSIONS

In this work we have proposed a comparative study of four different reconstruction methods. The comparison shows that the various reconstruction methods produce very different 3D models. Every methods has its own advantages and limitations; The FE and DMC appear to be close in terms of overall results and produce results with better standard deviation compared to the other two methods. The MPU is based on adaptive implicit functions which makes it more precise than the other three methods for many regions despite some extrapolations observed sometimes due to the complexity of the structure or noisy contours; it also requires to tune the tolerance parameter in order to produce visually plausible results. As for CI, if we consider smaller distances less than half a voxel size, it has the greater percentage of points however it has many points with important maximum distance value and generates sometimes non-precise reconstructed parts.

One of the possible extensions of this work would be to include metabolite quantification thanks to Magnetic Resonance Spectroscopy to improve the 3D reconstructed model. The final purpose is to have an accurate and precise 3D model that meets medical expectations and that can be used for diagnosis and virtual biopsy. In the long term, a precise reconstruction will offer a non-invasive technique to detect and localize the presence of a tumor, to classify its nature and to study the tumor metabolism in order to evaluate the response to possible therapies, thereby helping clinicians to optimize the treatment.

## ACKNOWLEDGEMENTS

This research work is funded by Region Nouvelle-Aquitaine, France (AAP ESR 2018 and 2019 Program) and SIEMENS Healthineers.

Loubna Lechelek, <http://orcid.org/0000-0003-1751-7762>

Sylvain Gerbaud, <http://orcid.org/0000-0002-2375-6754>

Rita Zrour, <http://orcid.org/0000-0002-0733-1326>

Mathieu Naudin, <http://orcid.org/0000-0001-8778-0297>

Carole Guillevin, <http://orcid.org/0000-0003-1075-4730>

Sebastien Horna, <http://orcid.org/0000-0002-9170-6513>

## REFERENCES

- [1] Bajaj, C.L.; Coyle, E.J.; Lin, K.N.: Arbitrary topology shape reconstruction from planar cross sections. *Graphical models and image processing*, 58(6), 524–543, 1996. <https://doi.org/10.1006/gmip.1996.0044>.
- [2] Boissonnat, J.D.: Shape reconstruction from planar cross sections. *Computer vision, graphics, and image processing*, 44(1), 1–29, 1988. [https://doi.org/10.1016/S0734-189X\(88\)80028-8](https://doi.org/10.1016/S0734-189X(88)80028-8).
- [3] Braude, I.; Marker, J.; Museth, K.; Nissanov, J.; Breen, D.: Contour-based surface reconstruction using mpu implicit models. *Graphical models*, 69(2), 2007, 139–157. <http://doi.org/10.1016/j.gmod.2006.09.007>.
- [4] Chernyaev, E.: Marching cubes 33: Construction of topologically correct isosurfaces. Tech. rep., 1995.
- [5] Custodio, L.; Etienne, T.; Pesco, S.; Silva, C.: Practical considerations on marching cubes 33 topological correctness. *Computers & graphics*, 37(7), 840–850, 2013. <https://doi.org/10.1016/j.cag.2013.04.004>.
- [6] Custodio, L.; Pesco, S.; Silva, C.: An extended triangulation to the marching cubes 33 algorithm. *Journal of the Brazilian Computer Society*, 25(1), 1–18, 2019. <https://doi.org/10.1186/s13173-019-0086-6>.
- [7] Dürst, M.J.: Re: Additional reference to "marching cubes". *ACM SIGGRAPH Computer Graphics*, 22(5), 243, 1988. <https://doi.org/10.1145/378267.378271>.
- [8] Fuchs, H.; Kedem, Z.M.; Uselton, S.P.: Optimal surface reconstruction from planar contours. *Communications of the ACM*, 20(10), 693–702, 1977. <https://doi.org/10.1145/359842.359846>.
- [9] Fujimura, K.; Kuo, E.: Shape reconstruction from contours using isotopic deformation. *Graphical Models and Image Processing*, 61(3), 127–147, 1999. <https://doi.org/10.1006/gmip.1999.0494>.
- [10] Galin, E.; Akkouche, S.: Fast surface reconstruction from contours using implicit surfaces. In *Implicit Surfaces*, vol. 98, 139–144, 1998.
- [11] Ganapathy, S.; Dennehy, T.G.: A new general triangulation method for planar contours. *ACM Siggraph Computer Graphics*, 16(3), 69–75, 1982. <https://doi.org/10.1145/965145.801264>.
- [12] Grosso, R.; Zint, D.: Parallel reconstruction of quad only meshes from volume data. vol. 1, 102–112. *VISIGRAPP'20, Valletta, MALTA*, 2020. <http://doi.org/10.5220/0008948701020112>.
- [13] Jones, M.W.; Chen, M.: A new approach to the construction of surfaces from contour data. In *Computer Graphics Forum*, vol. 13, 75–84. Wiley Online Library, 1994. <https://doi.org/10.1111/1467-8659.1330075>.
- [14] Levin, D.: Multidimensional reconstruction by set-valued approximations. *IMA Journal of Numerical Analysis*, 6(2), 173–184, 1986. <https://doi.org/10.1093/imanum/6.2.173>.
- [15] Lewiner, T.; Lopes, H.; Vieira, A.W.; Tavares, G.: Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, 8(2), 1–15, 2003. <https://doi.org/10.1080/10867651.2003.10487582>.
- [16] Lorensen, W.E.; Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4), 1987, 163–169. <http://doi.org/10.1145/37401.37422>.
- [17] Marker, J.; Braude, I.; Museth, K.; Breen, D.E.: Contour-based surface reconstruction using implicit curve fitting, and distance field filtering and interpolation. In *Volume Graphics*, vol. 2006, 1–9, 2006. <http://dx.doi.org/10.2312/VG/VG06/095-102>.
- [18] Meyers, D.; Skinner, S.; Sloan, K.: Surfaces from contours. *ACM Transactions On Graphics (TOG)*, 11(3), 228–258, 1992. <https://doi.org/10.1145/130881.131213>.

- [19] Natarajan, B.K.: On generating topologically consistent isosurfaces from uniform samples. *The Visual Computer*, 11(1), 52–62, 1994. <https://doi.org/10.1007/BF01900699>.
- [20] Nielson, G.: On marching cubes. *Visualization and Computer Graphics, IEEE Transactions on*, 9(4), 2003, 283–297. <http://doi.org/10.1109/TVCG.2003.1207437>.
- [21] Nielson, G.M.; Hamann, B.: The asymptotic decider: resolving the ambiguity in marching cubes. In *IEEE visualization*, vol. 91, 83–91, 1991. <https://doi.org/10.1109/VISUAL.1991.175782>.
- [22] Ohtake, Y.; Belyaev, A.; Alexa, M.; Turk, G.; Seidel, H.P.: Multi-level partition of unity implicits. In *Acm Siggraph 2005 Courses*, 173–es, 2005. <https://doi.org/10.1145/882262.882293>.
- [23] Park, H.; Kwon, M.J.; Han, Y.: Techniques in image segmentation and 3d visualization in brain mri and their applications. *Medical Imaging Systems Technology: Methods in cardiovascular and brain systems*, 5, 2005, 207. [http://doi.org/10.1142/9789812701046\\_0007](http://doi.org/10.1142/9789812701046_0007).
- [24] Pieper, S.; Halle, M.; Kikinis, R.: 3d slicer. vol. 1, 632–635. *IEEE International Symposium on Biomedical Imaging: Nano to Macro (ISBI)*, Arlington, VA, USA, 2004. <http://doi.org/10.1109/ISBI.2004.1398617>.
- [25] Pihuit, A.; Palombi, O.; Cani, M.P.: Reconstruction implicite de surfaces 3d à partir de régions 2d dans des plans parallèles, 2009.
- [26] Pinter, C.; Lasso, A.; Wang, A.; Jaffray, D.; Fichtinger, G.: Slicerrt, radiation therapy research toolkit for 3d slicer. *Med. Phys*, 39(10), 6332–6338, 2012. <https://doi.org/10.1118/1.4754659>.
- [27] Schroeder, W.; Maynard, R.; Geveci, B.: Flying edges: A high-performance scalable isocontouring algorithm. 33–40. *IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, Chicago, IL UNITED STATES, 2015. <http://doi.org/10.13140/RG.2.1.3415.9609>.
- [28] Schroeder, W.J.; Lorensen, B.; Martin, K.: *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware, 2004.
- [29] Sunderland, K.; Woo, B.; Pinter, C.; Fichtinger, G.: Reconstruction of surfaces from planar contours through contour interpolation. vol. 9415, 435–442. *International Society for Optics and Photonics*, Orlando, FLORIDA, UNITED STATES, 2015. <http://doi.org/10.1117/12.2081436>.
- [30] Taubin, G.; Zhang, T.; Golub, G.: Optimal surface smoothing as filter design. In *European Conference on Computer Vision*, 283–292. Springer, 1996. <http://dx.doi.org/10.1007/BFb0015544>.
- [31] Yoo, D.J.: Three-dimensional surface reconstruction of human bone using a b-spline based interpolation approach. *Computer-Aided Design*, 43(8), 934–947, 2011. <https://doi.org/10.1016/j.cad.2011.03.002>.