HAL
open science

On Volumetric Shape Reconstruction from Implicit Forms

Li Wang, Franck Hétroy-Wheeler, Edmond Boyer

To cite this version:

Li Wang, Franck Hétroy-Wheeler, Edmond Boyer. On Volumetric Shape Reconstruction from Implicit Forms. ECCV 2016 - European Conference on Computer Vision, Oct 2016, Amsterdam, Netherlands. 10.1007/978-3-319-46487-9_11 . hal-01349059

HAL Id: hal-01349059

https://inria.hal.science/hal-01349059

Submitted on 16 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Volumetric Shape Reconstruction from Implicit Forms

Li Wang & Franck Hétroy-Wheeler & Edmond Boyer

Univ. Grenoble Alpes & Inria & CNRS, LJK, F-38000 Grenoble, France
{ li.wang, franck.hetroy, edmond.boyer }@inria.fr

Abstract. In this paper we report on the evaluation of volumetric shape reconstruction methods that consider as input implicit forms in 3D. Many visual applications build implicit representations of shapes that are converted into explicit shape representations using geometric tools such as the Marching Cubes algorithm. This is the case with image based reconstructions that produce point clouds from which implicit functions are computed, with for instance a Poisson reconstruction approach. While the Marching Cubes method is a versatile solution with proven efficiency, alternative solutions exist with different and complementary properties that are of interest for shape modeling. In this paper, we propose a novel strategy that builds on Centroidal Voronoi Tessellations (CVTs). These tessellations provide volumetric and surface representations with strong regularities in addition to provably more accurate approximations of the implicit forms considered. In order to compare the existing strategies, we present an extensive evaluation that analyzes various properties of the main strategies for implicit to explicit volumetric conversions: Marching cubes, Delaunay refinement and CVTs, including accuracy and shape quality of the resulting shape mesh.

1 Introduction

Visual computing applications usually consider explicit representations for 3D shapes, in the form of surface or volume meshes in general. This is true for most visualization applications and also for applications that require local neighboring information within the shape or on its surface, as often the case with shape optimization or shape deformation applications for example. In order to generate such explicit representations from visual observations many methods consider as input an implicit representation that identifies the shape as being a region $\mathcal{V}$ within an observation domain $\Omega \in \mathbb{R}^3$. Such implicit representation is typically given as a scalar function $f : \Omega \to \mathbb{R}$ which takes different values inside and outside $\mathcal{V}$, for instance an indicator function or a distance function in 3D. These implicit representations f are often encountered in reconstruction applications that consider point clouds, as obtained with for instance stereo, multi-stereo and depth scanning apparatus, *e.g.* [1] [2] (see Fig. 1). They can also be built directly from image primitives, for instance the implicit visual hull form, *e.g.* [3] with image silhouettes. The conversion then from implicit to explicit representations

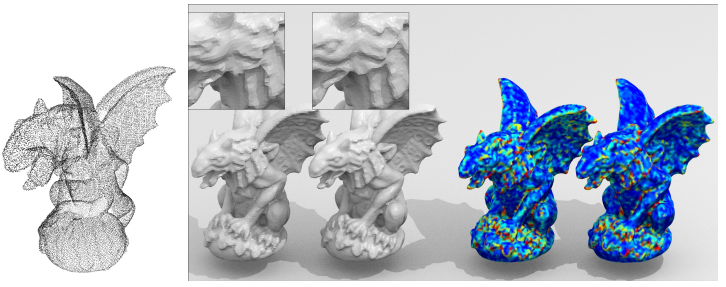


Fig. 1. The Gargoyle multi-view point cloud and the associated Poisson reconstructions with Marching Cubes and CVT. Distances to the implicit form are color encoded on the right, from low (blue) to high (red).

usually consists in the polyhedrization of the region $\mathcal{V}$, where both the resulting polyhedral volume and the associated polygonal surface approximations are potentially considered by vision and graphics applications *e.g.* [4] [5] [6]. In this paper we report on a novel approach to solve for this important conversion step and we provide a comparative study of the main strategies available.

Existing approaches for such 3D implicit form conversions into volumetric tessellations can be roughly divided into two categories. A first category, that includes the Marching Cubes method [7] and its extensions, adopts a fixed-grid strategy where the observation domain Ω is discretized into cells that are traditionally cubic. Inside and outside cells are identified with respect to the input implicit function and the boundary cells can be further polygonized into a triangle mesh approximating the shape surface. This strategy is efficient and fast and has been very widely used in vision and graphics applications over the last decades. However the 3D shape discretization into cubic cells with constrained orientations produces a poor shape tessellation which can result in surface approximations with elongated or small triangles. Thus, an additional re-meshing step is consequently often required. In addition, attaching the grid to Ω makes the tessellation changing with any shape transformation, even rigid. A second category of approaches, such as [8] with the Delaunay tetrahedrization, discretizes instead the inside region $\mathcal{V}$. These approaches usually provide better shape tessellations which are as well independent of rigid shape transformations and hence plausibly better suited for dynamic scene modeling. Still, they require expert control to monitor the cell refinement step that is performed. Moreover, as the boundary of a tetrahedral structure can present non manifold parts it is difficult to guarantee a correct topology for the boundary mesh approximating the surface. In this paper, we explore a different strategy that also belongs to the second category and discretizes $\mathcal{V}$ instead of Ω. The approach builds on Centroidal Voronoi Tessellations (CVTs) that provide regular shape tessellations which boundaries are obtained by clipping frontier cells with the given implicit boundary form. In contrast to Delaunay-based methods, the boundary surface

of the output volume is, by construction, manifold and the approach has only a few parameters.

In order to highlight the main features of the proposed CVT approach, including its limitations, we propose in this paper a comprehensive evaluation that compares all methods with respect to consistent criteria: the topology correctness of the produced volumetric model; The accuracy with respect to the input implicit form; The quality of the volumetric tessellation; And the computation times. The main result of this evaluation is that the CVT approach provably outperforms the other strategies in terms of accuracy and cell regularity, however to the price of higher computation times with respect to Marching Cubes. To summarize, our main contribution is twofold:

1. We introduce a CVT approach to polyhedrize implicit shape representations and provide a practical algorithm for that purpose.
2. We present a qualitative and quantitative comparison of Marching Cubes-, Delaunay- and CVT-based strategies to produce volume and surface shape approximations.

These contributions are detailed in Sections 3 and 4 respectively, after a presentation of related works in Section 2.

2 Related Work

As mentioned earlier methods for the polyhedrization of implicit forms fall into two main categories: first, *Eulerian* methods that consider a grid discretizing the observation domain Ω and second *Lagrangian* methods which perform a discretization of the shape volume $\mathcal{V}$.

Eulerian strategies. Originally designed for isosurface extraction, *i.e.* estimating the implicit surface defined by $\{x \in \Omega, f(x) = cst\}$, the Marching cubes (MC) algorithm introduced in [7] is the most prominent approach in this category as a result of its efficiency and versatility of applications. It considers a regular cubical grid that partitions Ω, where Ω is typically a bounding box. For each cube within the grid that intersects the shape, the algorithm determines the intersection faces by linearly interpolating, along cube edges, the f values at the cube vertices. Many methods have then been proposed that adapt this strategy to sharp features, *e.g.* [9], or to resolve topological ambiguities induced by the original method, see [10] for a survey. Other methods have also proposed more complex interpolation schemes [11] [12] to better locate surface points along cube edges. To speed up computations, which may be slow since the complexity is cubic with respect to the grid size, several authors have suggested to replace the grid by an adaptive structure such as an octree, *e.g.* [13]. Marching Cubes methods focus on the polygonization of the shape boundary surface and not on the volumetric tessellations which is irregular by construction: interior cells are cubic while boundary cells are not. Extensions to other subdivision schemes

including tetrahedral, octahedral and hexahedral subdivisions, *e.g.* [14] [15] [16] have also been proposed that provide more isotropic polyhedral cells. However the discretization grid in these schemes is still fixed.

Lagrangian strategies. Instead of subdividing Ω, approaches in this category tessellate the shape volume. The interest is first to reduce the complexity, which allows for better accuracies than MC at similar resolutions. This can be an important feature when modeling large scenes, *e.g.* [2]. Second, attaching the subdivision to the shape eases the implementation of kinematic models that can be defined over volumetric representations. This can help modeling dynamic scenes as in [5]. In this category, shape tetrahedrization is widely used to generate a volumetric tessellation. For example, the isosurface stuffing algorithm from [17] creates a regular tetrahedrization from a body-centered cubic (BCC) grid. As for the Marching cubes approaches described above, the resulting tessellation depends however on the orientation of the grid. The Delaunay refinement technique, as in [8], also generates Delaunay tetrahedrizations with guarantees on tetrahedron shapes. However, degenerate tetrahedra, typically slivers, can still appear, although their number can be reduced by global [18] or local [19] optimization techniques.

Centroidal Voronoi Tessellations (CVTs) are a special type of Voronoi tessellations with regular Voronoi cells [20]. Such tessellations are known to be optimal quantizers [21] and their cells, mostly truncated octahedra, are more isotropic than cubes or tetrahedra. Consequently, CVTs have been used to discretize 2D and 3D shapes in many scientific domains [22]. While methods have been proposed to clip a CVT to a surface mesh [20] [23] [24], to the best of our knowledge, none is able yet to handle implicit forms. We introduce therefore in the next section a new clipping method for CVTs.

Surface reconstruction. We focus in this paper on methods for the polyhedrization of implicit forms however it is worth mentioning approaches that reconstruct the zero-set surface of an implicit form, see *e.g.* [25] for a review. Approaches in this category focus on surface tessellation where we consider volumetric tessellations.

Evaluation. In the literature, 2D and 3D shape reconstruction techniques are mostly evaluated according to the *quality* of the constructed cells. This is indeed crucial for applications such as Finite Element Modeling. Quality is usually defined with respect to the shape of the cell [26]. In some cases the quality metrics reflect the fact that the cell should stay away from degenerate configurations. For instance, tetrahedra with at least one small angle are usually to be avoided [27]. In other cases, an ideal shape is defined, and the quality metrics are defined as distance to this ideal. This is the case with CVTs, for which the ideal cell shape is known to be a truncated octahedron as mentioned in [21]. In this work, the dimensionless second moment of a polytope is introduced and can be used as a

measure of the regularity of a CVT cell [20]. Metrics have also been proposed for other types of cells, such as hexahedra [26], as well as algebraically for general cells [28].

A few works have investigated the geometric accuracy of a given reconstruction. Geometric accuracy can be defined as a distance between the boundaries of the given input form and of the volumetric reconstruction obtained. *Metro* [29], introduced in the context of mesh simplification, is a common tool to evaluate the distance between two triangulated surfaces. More recently and focusing on surface reconstruction, Berger et al. [30] have recently introduced better metrics based on discrete differential geometry concepts in order to quantitatively evaluate distances between an implicit surface and a surface mesh. We build on this work for accuracy evaluation.

In Section 4, we compare the shape tessellations obtained with Marching Cubes, Delaunay refinement and CVT approaches using both shape quality and geometric accuracy criteria. We also discuss the theoretical guarantees given by each approach, as well as their computation times.

3 CVT for Implicit Forms

Centroidal Voronoi Tessellations are used in shape modeling to discretize 2D and 3D shapes into polygonal and polyhedral cells, respectively, centered around points called *sites* and distributed inside the shapes. CVT cells optimally partition the input domain in the sense of k-means clusters minimizing a variance or quantification error [22]. To this purpose, CVT algorithms alternatively re-estimate site locations and their associated cells, in an iterative manner. Approaches exist that compute CVTs given explicit forms for shapes, usually meshes as in [20] [23] [24]. We consider here the case of shapes defined by implicit forms in 3D and also, more specifically, implicit forms obtained from point clouds, a frequent case when modeling shapes with visual observations.

3.1 Background

Given a finite set of n points $X = \{x_i\}_{i=1}^n$ in $\mathbb{R}^3$, called *sites*, the *Voronoi cell* or *Voronoi region* Ω_i of x_i is defined as follows:

$$\Omega_i = \{x \in \mathbb{R}^3 \mid \|x - x_i\| \leq \|x - x_j\|, \ \forall j \neq i\},$$

where $\|.\|$ denotes the Euclidean distance. The partition of $\mathbb{R}^3$ into Voronoi cells is called a *Voronoi tessellation*. A *clipped Voronoi tessellation* is then the intersection between the Voronoi tessellation and a volume $\mathcal{V} \in \mathbb{R}^3$ and a *centroidal Voronoi tessellation* (CVT) is a special type of clipped Voronoi tessellation where the site of each Voronoi cell Ω_i is also its center of mass or centroid, *i.e.* :

$$x_i = \frac{\int_{\Omega_i} x \, \mathrm{d}x}{\int_{\Omega_i} \mathrm{d}x}.$$

This property ensures that CVTs are local minima of the quantization error $E : \mathbb{R}^{3n} \to \mathbb{R}$ below, also called the CVT energy function or the distortion [22].

$$E(X) = \sum_{i=1}^{n} E_i(X) = \sum_{i=1}^{n} \int_{\Omega_i} \|x - x_i\|^2 \, \mathrm{d}x. \tag{1}$$

3.2 Algorithm overview

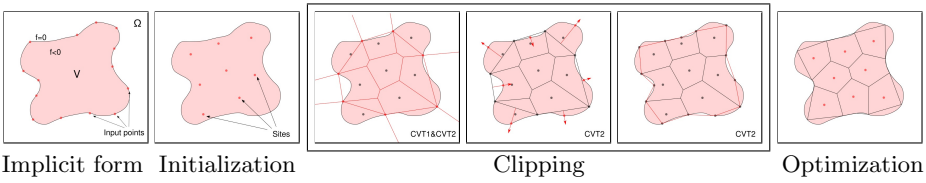
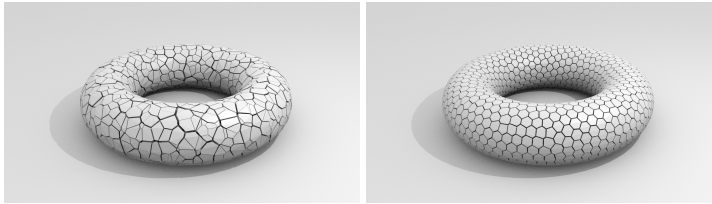


Fig. 2. The different steps of the CVT algorithm. The clipping and optimization steps are iterated until the sites are stabilized.

Our CVT algorithm considers as input an implicit function $f : \Omega \to \mathbb{R}$ defined over a domain $\Omega \in \mathbb{R}^3$ and such that $f(x) = 0$ on the boundary surface $\mathcal{S}$ of a shape $\mathcal{V}$. In the case of an indicator function, f has zero values outside $\mathcal{V}$ and the value 1 inside. The algorithm takes also as input the number of sites-cells n and follows the traditional CVT scheme below:

1. Initialization: find initial positions for the n sites inside $\mathcal{V}$.
2. Clipping: compute the Voronoi tessellation of the sites, then restrict it to $\mathcal{V}$ by computing its intersection with $\mathcal{S}$.
3. Optimization: update the position of the sites by minimizing the CVT energy function (1).

Steps 2 and 3 are iterated several times where the number of iterations is a user-defined parameter. Any initialization can, in principle, be applied here. In our experiments, the sites are randomly positioned inside $\mathcal{V}$. At the first iteration, the cells of the clipped Voronoi tessellation constructed after step 2 are not uniform nor regular (see Fig. 3 (a)) and do not minimize the CVT energy (1). Step 3 optimizes therefore the site locations in order to minimize E. In the literature, the two main strategies for such minimization are Lloyd's gradient descent method and the L-BFGS quasi-Newton method [31]. In our approach, we choose the latter since it is known to be faster [31]. As shown in Fig. 3 (b), once convergence is reached, the clipped Voronoi cells are almost uniform and regularly spaced. Besides, they yield a better approximation of the shape $\mathcal{V}$, as shown in Section 4.



(a) (b)

Fig. 3. Voronoi tessellations of a torus with and without optimization. (a) A clipped Voronoi tessellation with random initial positions for the sites. (b) Clipped CVT after optimization.

3.3 Clipping

In order to clip a Voronoi tessellation with the implicit function f describing the shape $\mathcal{V}$, we introduce an algorithm that consists of the following main steps (see Fig. 2):

1. Given the unbounded Voronoi tessellation $\bigcup_i \Omega_i$ of the sites $\{x_i\}$, identify which cells Ω_i intersect the implicit surface $\mathcal{S}$ bounding $\mathcal{V}$.
2. For each of these boundary cells Ω_j,
   (a) Compute the intersection between the edges of Ω_j and $\mathcal{S}$, this intersection being represented as a set of points P_j (red dots in Fig. 2 CVT1&CVT2).
   (b) Build the boundary clipped Voronoi cell Ω'_j as the convex hull of the intersection points in P_j and the vertices of Ω_j that are inside $\mathcal{V}$ (CVT1 in Fig. 2 CVT1&CVT2).
   (c) (CVT-2 only) Add to each boundary cell the point at the intersection between $\mathcal{S}$ and the ray along the normal to Ω'_j. Rebuild Ω'_j by connecting it with other intersection points of P_j (CVT2 in Fig. 2).

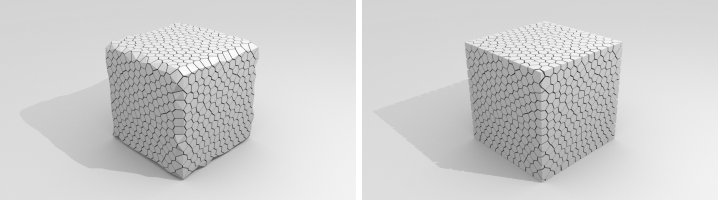
Step 1 is carried out by first converting infinite Voronoi cells into finite cells using a bounding surface around $\mathcal{V}$ and second by detecting boundary cells as cells with at least one vertex outside $\mathcal{V}$. Step 2 is discussed below.

Intersections of Ω_j with $\mathcal{S}$. Boundary cells are convex polytopes composed of bounded polygons and segments. We first interpolate f values along segments to find their intersections with f. To this purpose, several strategies can be considered depending on the information available on f. When both function values and derivatives are available, Hermite interpolation can be used, as advocated in [12]. It provides fast and accurate interpolated values as long as the local approximation is valid. When only function values are available, linear interpolation, with for instance the false position algorithm, can be used to iteratively locate the intersection. In any cases, the bisection method can be applied. It is slower than the previous strategies but more robust. A combination of the bisection and Hermite methods can also be considered to first reduce the search space

so that the Hermite approximation becomes more valid. Note that polyhedrization approaches are independent of the interpolation scheme and can all consider any of them. For the purpose of evaluation, and without loss of generality, we use the bisection method with all approaches in the comparisons presented in the evaluation section 4.

Convex hull. Once the edge intersection points are determined, the clipped Voronoi cell is computed as the convex hull of the intersection points and the cell vertices inside the surface. Since Voronoi cells are convex, this is guaranteed to provide a boundary surface with a correct topology (see CVT1&CVT2 in Fig. 2).

Sharp features. Sharp features, in case they occur, are not preserved by default by the clipping algorithm CVT1 (see Fig. 4 (a) for instance). This also true for Marching Cubes and Delaunay strategies for which specific solutions have been proposed, *e.g.* [9] and [32] respectively. CVT easily adapts to sharp features when identified as a list of points that can be simply assigned to their closest sites before the convex hull computation. This is a nice feature of the CVT strategy that is flexible and allows for such additional points without significantly increasing the complexity (only slightly modifying the convex hull computation) and while keeping the topological guarantees. This would be difficult to implement with the Marching Cubes or Delaunay strategies without fundamentally modifying the associated algorithms. Fig. 4 (b) shows an example of such a reconstruction.



(a) (b)

Fig. 4. Tessellations of the characteristic function of a cube: (a) CVT1 algorithm without additional points. (b) CVT2 algorithm with sharp feature points added.

4 Evaluation

In order to compare the different strategies for shape modeling mentioned different criteria can be considered. As emphasized in [33] for a similar evaluation, a given method will easily favor one of these criteria at the cost of the others, depending on the targeted application. In this work, our target is the reconstruction from implicit functions. This includes theoretical guarantees, the accuracy of the approximation with respect to the input implicit boundary surface, the quality of the resulting cells, and the time complexity.

4.1 Methodology

Methods. The four different methods we compare are the following:
- MC: the implementation of the Marching Cubes algorithm with topological guarantees [34];
- CGAL: the CGAL [35] implementation of the Delaunay tetrahedrization based algorithm, which uses the Delaunay refinement technique followed by mesh optimization to remove degenerated tetrahedra [36];
- CVT1[1]: a first proposed implementation of the clipped CVT algorithm, *i.e.* only Voronoi edge intersections with the surface are considered as surface points (see Section 3.3);
- CVT2[1]: an extension of CVT1 where an additional point which is the intersection between the surface and a ray estimated by boundary clipped Voronoi cell is added. (see Section 3.3).

To compare shape tessellations on a fair basis, similar resolutions, *i.e.* cell numbers, are required. While the resolution is easily imposed with CVT, which can be an advantage when similar shape discretizations are required, it is less easy with MC and space discretizations. In practice, we first compute the Marching Cubes tessellation. We then use $\sqrt{3}/2$ times the length of a cube as the targeted radius of a tetrahedron's circumsphere for the Delaunay tetrahedrization approach, in order for the size of this circumsphere to be similar to the size of the cube's circumsphere. For the CVT approaches, we simply sample randomly as many sites as the number of cubes inside the shape.

Data. Methods are first evaluated on a set of 97 object meshes, from the Princeton Segmentation Benchmark [37], for statistical comparisons on the accuracy (see Fig. 6). We next consider point clouds obtained from vision reconstructions and from which implicit forms are built. The Dancer (Fig. 5) was obtained with a multiview system followed by a Poisson implicit function estimation [35]. Gargoyle (Fig. 1) was obtained with [30] and the three other shapes, KneelingLady, Aquarius and Skull, were obtained with [1] followed by the same Poisson estimation [35] (other implicit function estimation could be considered). Dancer's MC results are extracted from 50 x 50 x 50 and 100 x 100 x 100 voxel grid for different resolution tests. Others' MC results are from 100 x 100 x 100 voxel grid.

4.2 Theoretical guarantees

Two theoretical guarantees are in practice often required: the manifoldness of the output volumetric mesh and the topological correctness with respect to the input form. A k-manifold is a k-dimensional object which is locally homeomorphic to a k-dimensional disk. A k-manifold allows for non ambiguous definitions of geometrical quantities such as the local geodesic neighborhood of any point, which is critical in many shape processing applications. It allows for instance to

[1] Source code will be released.

smooth and deform shapes in a consistent way. Topological correctness is the fact that input and output shapes present the same topology, for example the same number of components. This is an important property when considering properties over sets of shapes.

The original Marching Cubes algorithm [7] is known to produce surface meshes which can be non-manifold and which topology can differ from the input implicit surface. Many methods have been proposed to solve for non-manifoldness and topological ambiguities, see [10] for a survey. 2D Delaunay triangulations are known to be, under some sampling assumptions, good geometrical and topological approximations of the input shape [38]. This has been used for instance in order to robustly model surfaces evolving over time [39]. However, this is not the case in higher dimensions [40]. Thus, an additional post-processing step, such as [41], is required to guarantee manifoldness in the case of Delaunay tetrahedrizations. In contrast, a 3D CVT is manifold by construction: it is composed of convex 3D cells and on its surface every edge which is generated by the intersection of a bisector of CVT and the implicit surface is shared by exactly 2 faces because it belongs to a bisector.

4.3 Accuracy

The accuracy measures how close the estimated shape approximation is to the implicit form. To this aim we compare shape surfaces. The geometric similarity between two surfaces can be defined in several ways. Following the evaluation in [30], we consider distances between shapes in both directions and the following metrics:
- $Dmean$: the mean of distances between the input shape and the reconstructed surface in both directions;
- $Drms$: the root-mean-square (RMS) of these distances;
- $Nmean$: the mean angle deviation;
- $Nrms$: the RMS angle deviation.

Distances are estimated at points regularly distributed on both shapes and by searching for the closest point on the other shape in the facet normal directions. The same principle applies for normal angles that are estimated between closest points in the evaluation sets on both the input and the reconstructed surfaces. In order to build regular evaluation point sets, we use a particle system to sample implicit surfaces [30], and we compute 2D CVTs on the reconstructed boundary surface to generate sample points regularly distributed on both surfaces. Using a particle system and 2D CVTs to regularly distribute evaluation points on the surface ensures that we do not estimate distances between two discretizations of the input implicit surface at different scales, as can be the case with *Metro* [29].

4.4 Shape Quality

Besides the accuracy of the approximation, tessellations can also be compared with respect to the cell shape properties. Compactness, that ensures regularity, is for instance desirable for, *e.g.* , local discrete operations on shapes such as

deformations or quantification optimality. We first assess cell regularity for the Marching Cubes and CVT approaches (Delaunay tetrahedra are not compact by construction). For Delaunay tetrahedrizations, we compare them to CVTs using the dual tetrahedrizations of CVTs and boundary triangle quality metrics. Dual tetrahedrizations are computed by projecting the sites of the boundary Voronoi cells on the surface and then optimizing their position as in [23].

Cell regularity. Marching Cubes-generated tessellations are mostly made of cubes, however the boundary cells may be very irregular. In order to assess cell regularity, following [20] we use the criterion G_3 for polytopes referred to as the *dimensionless second moment* of the cell by Conway and Sloane [21]. The definition of G_3 for a cell Ω is:

$$G_3(\Omega) = \frac{\int_\Omega \|x - \hat{x}\|^2 \, \mathrm{d}x}{3Vol(\Omega)^{5/3}}$$

where $\hat{x}$ is the centroid of the cell and $Vol(\Omega)$ its volume. $G_3(\Omega)$ reflects how far Ω is from the optimal quantizer in three dimensions. Hence, this criterion intuitively accounts for the compactness of the cell.

Tetrahedron quality. We use four standard quality metrics for tetrahedra [23]:
- VQ_1: minimum dihedral angle of the tetrahedron;
- VQ_2: maximum dihedral angle;
- VQ_3: radius-ratio, defined as $\frac{3r_i}{r_c}$, where r_i and r_c are the radii of the inscribed and circumscribed spheres, respectively;
- VQ_4: meshing quality, defined as $\frac{12\sqrt[3]{9V^2}}{\sum l_{i,j}^2}$, where V is the volume of the tetrahedron and $l_{i,j}$ is the length of an edge i, j of the tetrahedron.

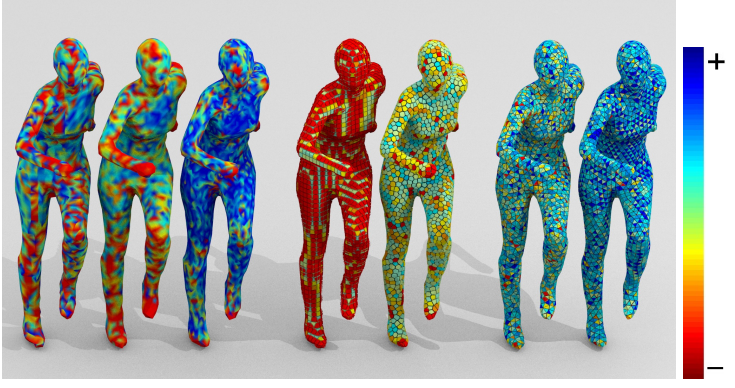
Boundary triangle quality. Four similar criteria are used for boundary triangles:
- SQ_1: minimum angle of the triangle;
- SQ_2: maximum angle;
- SQ_3: radius-ratio, defined as $\frac{2r_i}{r_c}$, where r_i and r_c are the radii of the inscribed and circumscribed circles, respectively;
- SQ_4: meshing quality, defined as $\frac{4\sqrt{3S}}{\sum l_{i,j}^2}$ where S is the area of the triangle and $l_{i,j}$ is the length of an edge i, j of the triangle.

4.5 Results and discussion

Accuracy. We first compare methods on a set of various shapes: 97 meshes taken from the Princeton Benchmark [37]. These meshes belong to 5 different categories and the algorithms have been run with a fixed resolution for a given category, which can however vary over categories: 50^3, 80^3 and 100^3 for the Marching Cubes. The number of sites for CVT is taken as the number of inner

cubes with MC, this for each mesh. Results in Fig. 6 show method rankings with respect the accuracy criteria defined previously. It demonstrates that CVT approaches are statistically significantly better than MC and Delaunay.

Accuracy quantitative have also been conducted on the mentioned vision datasets (see Table 2). Additionally, qualitative results are color-coded in Fig. 1 and Fig. 5. In these experiments, 10 iterations are performed for both Delaunay tetrahedrization and CVT approaches. CGAL failed to compute a tetrahedrization on the Gargoyle and Skull datasets (the process was stopped after 18 hours of computation). These results show that CVT2 performs better than other approaches on all our experiments. On point datasets that describe smooth shapes (Dancer and Gargoyle), CVT1 performs better than Marching Cubes even without the optimization step.



(MC) (Del) (CVT1) (MC) (CVT1) (Del) (CVT1)

Fig. 5. Accuracy (left), cell regularity (middle) and tetrahedron quality (right) of 3D implicit form tessellations with MC, Delaunay refinement (Del) and CVT1.

Shape quality. Table 2 shows the maximum *Gmax* and the mean *Gmean* values of the cell regularity $G_3(\Omega)$ over all cells of Marching Cubes and CVTs tessellations. These results show the benefit of optimizing site positions to generate more regular cells. The mean tetrahedron and boundary triangle quality measures are given in Table 3. According to our experiments, the CVT approach gives better results than the Delaunay tetrahedrization for all of them.

Computation times. Timings are shown in Table 1. CGAL fails to compute a tetrahedrization for the Gargoyle and the Skull datasets. The Marching Cubes algorithm is the fastest of the four tested methods, as a result of the increased complexity of Delaunay and Voronoi tessellation computations. It has

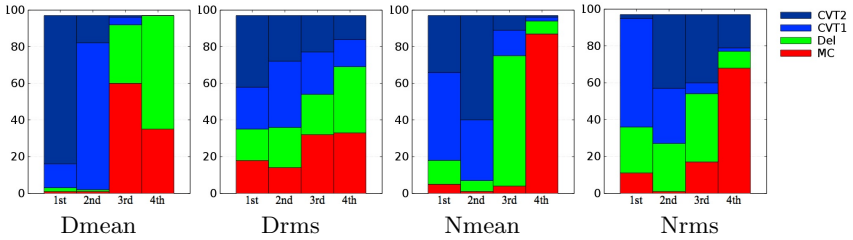


Fig. 6. Accuracy rankings on 100 meshes from the Princeton Benchmark [37]. Implicit forms were obtained using Poisson reconstructions [35] and accuracies measured on samples obtained using the particle system approach [30].

to be mentioned that most of the computation time of CVT approach lies in the 3D Delaunay triangulation (dual of CVT) which is costly and cannot be avoided (it takes about 70% of the CVT computation time en average). The CVT approach performs anyway always faster than the Delaunay tetrahedrization.

Method	DancerLow	Dancer	Aquarius	KneelingLady	Gargoyle	Skull
MC	0.1	0.6	1.6	1.1	1.3	1.9
CGAL (10 it.)	1036	1052	162	198	-	-
CVT1 (0 it.)	0.4	1.4	7.2	7.4	9.1	18
CVT1 (10 it.)	4.9	25	135	123	160	342
CVT2 (10 it.)	5.1	25	136	125	165	363

Table 1. Computational time (s) for each experiment.

5 Conclusion

We have presented a comparison of different strategies to convert an implicit form in 3D into an explicit shape representation. This includes a new strategy that uses Centroidal Voronoi Tessellations to build a 3D mesh composed of convex Voronoi cells around a pre-defined number of sites. The evaluation shows that CVTs provide the most accurate and the most regular shape tessellations when compared to Marching Cubes and Delaunay refinement strategies. As such, CVTs are a good alternative to Marching cubes, in particular when modeling dynamic scenes for which accuracy, regularity and invariance to rigid transformation are desirable properties of shape tessellation. Delaunay refinement, while providing boundary surfaces with *good* triangles does not outperform CVTs nor Marching cubes in any case. Finally, Marching cubes is always the fastest strategy, hence a solution for applications requiring fast solutions, though less accurate and regular than CVTs.

Dataset (nb of sites)	Method	Dmean	Drms	Nmean	Nrms	Gmax	Gmean	CVT2
Dancer Low (2365)	MC	27.16	29.72	1.31	1.41	3210	19.30	
	CGAL (10 it.)	27.12	29.72	1.30	1.41	-	-	
	CVT1 (0 it.)	27.14	29.67	1.31	1.41	1647	11.00	
	CVT1 (10 it.)	27.13	29.66	1.30	1.41	11.84	8.18	
	CVT2 (10 it.)	27.05	29.64	1.30	1.41	11.84	8.15	
Dancer (14474)	MC	22.10	25.46	1.14	1.25	5999	14.49	
	CGAL (10 it.)	21.72	25.11	1.14	1.25	-	-	
	CVT1 (0 it.)	21.86	25.41	1.13	1.25	75.85	9.75	
	CVT1 (10 it.)	21.76	25.25	1.12	1.24	9.82	8.07	
	CVT2 (10 it.)	21.16	25.29	1.12	1.24	9.82	8.03	
Aquarius (64588)	MC	21.64	23.99	1.32	1.42	29825	12.17	
	CGAL (10 it.)	21.65	23.99	1.32	1.42	-	-	
	CVT1 (10 it.)	21.83	24.14	1.33	1.42	13.17	8.16	
	CVT2 (10 it.)	21.59	23.94	1.31	1.41	13.02	8.15	
Kneeling Lady (73150)	MC	2.55	3.98	0.40	0.63	5870	11.80	
	CGAL (10 it.)	2.60	3.83	0.40	0.63	-	-	
	CVT1 (10 it.)	2.58	4.00	0.40	0.63	19.05	8.16	
	CVT2 (10 it.)	2.50	3.93	0.39	0.62	19.05	8.15	
Gargoyle (106497)	MC	0.38	0.54	0.20	0.29	54467	12.65	
	CVT1 (10 it.)	0.38	0.54	0.19	0.28	10.20	8.02	
	CVT2 (10 it.)	0.37	0.53	0.19	0.28	10.19	8.00	
Skull (225034)	MC	7.12	9.76	0.82	1.00	82716	11.26	
	CVT1 (10 it.)	7.15	9.81	0.83	1.00	17.87	8.07	
	CVT2 (10 it.)	7.07	9.67	0.82	1.00	17.56	8.06	

Table 2. Accuracy and regularity results on the datasets. (Dmean, Drms, Gmax, Gmean) $\times 10^{-2}$.

Dataset	Method	$\overline{VQ_1}$	$\overline{VQ_2}$	$\overline{VQ_3}$	$\overline{VQ_4}$	$\overline{SQ_1}$	$\overline{SQ_2}$	$\overline{SQ_3}$	$\overline{SQ_4}$
Dancer	CGAL (10 it.)	50.75	96.48	0.84	0.90	47.92	75.13	0.93	0.94
	CVT1 (10 it.)	**51.32**	**95.49**	**0.85**	**0.91**	**52.56**	**69.23**	**0.97**	**0.97**
Aquarius	CGAL (10 it.)	51.35	95.89	0.83	0.90	47.73	75.37	0.92	0.94
	CVT1 (10 it.)	**51.55**	**95.33**	**0.84**	**0.92**	**52.56**	**69.23**	**0.97**	**0.97**
Kneeling Lady	CGAL (10 it.)	51.54	95.71	0.83	0.90	48.32	74.59	0.93	0.94
	CVT1 (10 it.)	**51.63**	**95.16**	**0.84**	**0.92**	**50.95**	**71.27**	**0.96**	**0.96**

Table 3. Mean tetrahedron quality measures over all cells of the tessellation and mean triangle quality measures over all boundary triangles of the tessellation (best result in bold).

References

1. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE transactions on pattern analysis and machine intelligence **32**(8) (2010) 1362–1376 1, 9
2. Labatut, P., Pons, J.P., Keriven, R.: Robust and efficient surface reconstruction from range data. In: Computer graphics forum. Volume 28., Wiley Online Library (2009) 2275–2290 1, 4
3. Esteban, C.H., Schmitt, F.: Silhouette and stereo fusion for 3d object modeling. Computer Vision and Image Understanding **96**(3) (2004) 367–392 1
4. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2015) 1912–1920 2
5. Allain, B., Franco, J.S., Boyer, E.: An efficient volumetric framework for shape tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2015) 268–276 2, 4
6. Huang, C.H., Allain, B., Franco, J.S., Navab, N., Ilic, S., Boyer, E.: Volumetric 3d tracking by detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2016) 2
7. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: ACM siggraph computer graphics. Volume 21., ACM (1987) 163–169 2, 3, 10
8. Shewchuk, J.R.: Tetrahedral mesh generation by delaunay refinement. In: Proceedings of the fourteenth annual symposium on Computational geometry, ACM (1998) 86–95 2, 4
9. Kobbelt, L.P., Botsch, M., Schwanecke, U., Seidel, H.P.: Feature sensitive surface extraction from volume data. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM (2001) 57–66 3, 8
10. Newman, T.S., Yi, H.: A survey of the marching cubes algorithm. Computers & Graphics **30**(5) (2006) 854–879 3, 10
11. Lempitsky, V.: Surface extraction from binary volumes with higher-order smoothness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 1197–1204 3
12. Fuhrmann, S., Kazhdan, M., Goesele, M.: Accurate isosurface interpolation with hermite data. In: 3D Vision (3DV), 2015 International Conference on, IEEE (2015) 256–263 3, 7
13. Kazhdan, M., Klein, A., Dalal, K., Hoppe, H.: Unconstrained isosurface extraction on arbitrary octrees. In: Symposium on Geometry Processing. Volume 7. (2007) 256–263 3
14. Chan, S.L., Purisima, E.O.: A new tetrahedral tesselation scheme for isosurface generation. Computers & Graphics **22**(1) (1998) 83–90 4
15. Carr, H., Theußl, T., Möller, T.: Isosurfaces on optimal regular samples. In: ACM International Conference Proceeding Series. Volume 40., Citeseer (2003) 39–48 4
16. Sinha, S.N., Mordohai, P., Pollefeys, M.: Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: 2007 IEEE 11th International Conference on Computer Vision, IEEE (2007) 1–8 4
17. Labelle, F., Shewchuk, J.R.: Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In: ACM Transactions on Graphics (TOG). Volume 26., ACM (2007) 57 4

18. Alliez, P., Cohen-Steiner, D., Yvinec, M., Desbrun, M.: Variational tetrahedral meshing. In: ACM Transactions on Graphics (TOG). Volume 24., ACM (2005) 617–625 4
19. Tournois, J., Wormser, C., Alliez, P., Desbrun, M.: Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. ACM Transactions on Graphics **28**(3) (2009) 75:1–75:9 4
20. Wang, L., Hétroy-Wheeler, F., Boyer, E.: A hierarchical approach for regular centroidal voronoi tessellations. In: Computer Graphics Forum. Volume 35., Wiley Online Library (2016) 152–165 4, 5, 11
21. Conway, J., Sloane, N.: Voronoi regions of lattices, second moments of polytopes, and quantization. IEEE Transactions on Information Theory **28**(2) (1982) 211–226 4, 11
22. Du, Q., Faber, V., Gunzburger, M.: Centroidal voronoi tessellations: applications and algorithms. SIAM review **41**(4) (1999) 637–676 4, 5, 6
23. Yan, D.M., Wang, W., Lévy, B., Liu, Y.: Efficient computation of clipped voronoi diagram for mesh generation. Computer-Aided Design **45**(4) (2013) 843–852 4, 5, 11
24. Lévy, B.: Restricted voronoi diagrams for (re)-meshing surfaces and volumes. In: 8th International Conference on Curves and Surfaces. Volume 6. (2014) 14 4, 5
25. de Araujo, B., Lopes, D.S., Jepp, P., Jorge, J.A., Wyvill, B.: A survey on implicit surface polygonization. ACM Computing Surveys (CSUR) **47**(4) (2015) 60 4
26. Field, D.A.: Qualitative measures for initial meshes. International Journal for Numerical Methods in Engineering **47**(4) (2000) 887–906 4, 5
27. Cheng, S.W., Dey, T.K., Edelsbrunner, H., Facello, M.A., Teng, S.H.: Silver exudation. Journal of the ACM (JACM) **47**(5) (2000) 883–904 4
28. Knupp, P.M.: Algebraic mesh quality metrics. SIAM journal on scientific computing **23**(1) (2001) 193–218 5
29. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: measuring error on simplified surfaces. In: Computer Graphics Forum. Volume 17., Wiley Online Library (1998) 167–174 5, 10
30. Berger, M., Levine, J.A., Nonato, L.G., Taubin, G., Silva, C.T.: A benchmark for surface reconstruction. ACM Transactions on Graphics (TOG) **32**(2) (2013) 20 5, 9, 10, 13
31. Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.M., Lu, L., Yang, C.: On centroidal voronoi tessellationenergy smoothness and fast computation. ACM Transactions on Graphics (ToG) **28**(4) (2009) 101 6
32. Cheng, S.W., Dey, T.K., Ramos, E.A.: Delaunay refinement for piecewise smooth complexes. Discrete & Computational Geometry **43**(1) (2010) 121–166 8
33. Meyer, M., Kirby, R.M., Whitaker, R.: Topology, accuracy, and quality of isosurface meshes using dynamic particles. IEEE Transactions on Visualization and Computer Graphics **13**(6) (2007) 1704–1711 8
34. Lewiner, T., Lopes, H., Vieira, A.W., Tavares, G.: Efficient implementation of marching cubes' cases with topological guarantees. Journal of graphics tools **8**(2) (2003) 1–15 9
35. CGAL: Computational Geometry Algorithms Library http://www.cgal.org/. 9, 13
36. Jamin, C., Alliez, P., Yvinec, M., Boissonnat, J.D.: Cgalmesh: a generic framework for delaunay mesh generation. ACM Transactions on Mathematical Software (TOMS) **41**(4) (2015) 23 9
37. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3d mesh segmentation. In: ACM Transactions on Graphics (TOG). Volume 28., ACM (2009) 73 9, 11, 13

38. Boissonnat, J.D., Oudot, S.: Provably good sampling and meshing of surfaces. Graphical Models **67**(5) (2005) 405–451 10
39. Pons, J.P., Boissonnat, J.D.: Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007) 1–8 10
40. Oudot, S.Y.: On the topology of the restricted delaunay triangulation and witness complex in higher dimensions. arXiv preprint arXiv:0803.1296 (2008) 10
41. Lhuillier, M.: 2-manifold tests for 3d delaunay triangulation-based surface reconstruction. Journal of Mathematical Imaging and Vision **51**(1) (2015) 98–105 10