Digital surface regularization with guarantees

David Coeurjolly*, Jacques-Olivier Lachaud† and Pierre Gueth‡
*Univ Lyon, CNRS, LIRIS, Lyon, France
†Univ Savoie Mont Blanc, LAMA, Chambéry, France
‡Adobe

Abstract—Voxel based modeling is a very attractive way to represent complex multi-material objects. Beside artistic choices of pixel/voxel arts, representing objects as voxels allows efficient and dynamic interactions with the scene. For geometry processing purposes, many applications in material sciences, medical imaging or numerical simulation rely on a regular partitioning of the space with labeled voxels. In this article, we consider a variational approach to reconstruct interfaces in multi-labeled digital images. This approach efficiently produces piecewise smooth quadrangulated surfaces with some theoretical stability guarantee. Non-manifold parts at intersecting interfaces are handled naturally by our model. We illustrate the strength of our tool for digital surface regularization, as well as voxel art regularization by transferring colorimetric information to regularized quads and computing isotropic geodesic on digital surfaces.

Index Terms—Digital Geometry, Surface Reconstruction, Voxel Geometry Processing

1 Introduction

In many aspects of Computer Graphics, the representation of complex objects as data attached to voxels becomes more and more familiar. A key feature of voxel-based modeling is that many geometry and topology processing tasks (collision detection, constructive solid geometry, ray-object intersection, homotopic transforms...) can be trivially defined or efficiently implemented within this model. Together with very efficient data structures [1]–[3], voxel grids are widely used to represent or simulate complex phenomena that are volumetric, multi-material and/or dynamic (*e.g.*[4]–[6]). In rendering, voxel-based global illumination techniques reformulate the illumination equation and take into account the spatial subdivision domain to pre-compute material and light/material information. They thus obtain a physically based rendering whose complexity does not depend anymore on the geometrical complexity of the objects in the scene but on the grid size (which can be adapted to the device viewport) [7]–[9]. Voxel art and low-resolution voxel modeling can also be observed in the game industry thanks to some dedicated modeling tools [10]. In the spirit of [11] which addresses the issue of depixelizing pixel art, our method transforms the crispy representation of voxel surfaces into a smooth reconstruction even for low-resolution objects.

Beside modeling tasks, geometry processing of regular labeled images is crucial in many material sciences or medical imaging applications (*e.g* [12]–[14]) as many acquisition devices generate regularly spaced volumetric data.

In this article, we consider a variational formulation, first introduced in [15], to efficiently reconstruct interfaces in multi-labeled 3D images. Starting from digital surfaces defined as boundaries of voxel sets, we construct a piecewise smooth quadrangulated surface with geometric and topological theoretical guarantees that captures the singularities of the input voxel geometry. As we maintain a mapping between original digital surface quads and regularized ones, we can transfer information (*e.g.* color, embossing or bump map) from the multi-labeled image to the regularized interfaces. Such smooth surfaces can either be used for visualization purposes, or as an intermediate representation for later geometry processing tasks.

2 Context and contributions

2.1 Related work

Iso-contouring approaches. From the seminal work of Lorensen and Cline [16], the Marching-Cubes (MC) approach is one of the most popular approach to extract a triangular iso-surface from a 3-D gray-level volumetric image. This approach locally constructs the triangulation from a set of canonical configurations with vertices obtained by interpolation from image values. Many extensions have been proposed (see [17] for a survey). For instance to fix topological ambiguities [18], to handle multiple labels or interfaces [19], to quickly generate adaptive mesh on GPU [20], [21], or to extract feature preserving surfaces [22]. Using a similar local construction idea, Dual-Contouring (DC) techniques construct an iso-surface from implicit functions (with positions and normal vector field as input) [23]. These approaches produce smooth surfaces, even in adaptive and multi-material cases, but require both a precise normal vector field at the iso-surface and precise positions on the surface. Again, many derived works have been proposed to control the overall surface topology [24] or to provide fast GPU implementations [25]. For voxel art reconstruction, we can also cite the work of Muniz *et al.* [26] which mainly focuses on colorimetric transfer between the voxel object and MC-like surfaces. Our approach follows a similar idea for color mapping using a robust regularization of the surfaces geometry.

Finally, on binary 3D images, the "shrouds" approach of [27] builds a smooth surface that separates interior from exterior voxels. This 3D problem is transformed into a series of 2D smooth contour reconstruction problems, by

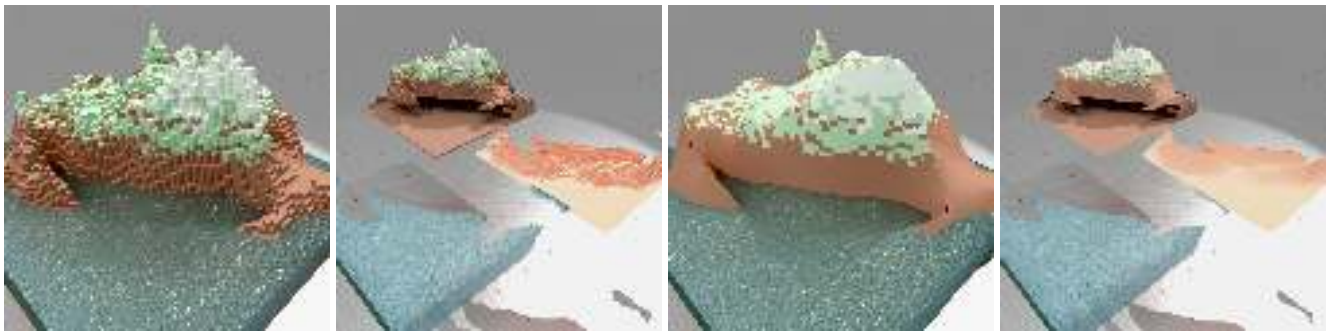


Fig. 1. **Reconstruction of the interfaces of a multi-labeled image**: *(from left to right)* input voxel art decomposed into two labels (sea, island, $64 \times 68 \times 40$) and three interfaces, piecewise smooth reconstruction with consistent interfaces between labels.

slicing the 3D volumes with axis-align planes. Each contour is regularized so as to minimize its squared curvature and the separation property is obtained by forcing each vertex to lie on its unit edge joining an interior voxel to an exterior voxel. Those 2D problems are intertwined since each vertex is shared by two slices. Since this method gives comparable results to our approach, we will compare them in the experiment section. Note nevertheless that this method is limited to closed surface reconstruction and its extension to open surfaces or multi-labeled image is not trivial.

Volumetric approaches. As we are dealing with volumetric multi-labeled images, solutions exist from volumetric meshing or volumetric reconstruction [14], [28]–[31]. These approaches aim at reconstructing a volumetric (usually tetrahedral) mesh which preserves multi-labeled interfaces, while optimizing the quality of geometrical elements, crucial when solving Finite Elements Method problems. Usually starting from structures like Delaunay triangulation, these techniques optimize the geometry using iterative or variational formulations. In this paper, we rather focus on the high quality of reconstructed interfaces within the multi-labeled dataset while keeping a mapping between input and output surface elements. We compare our method with some of these techniques.

Regularization of surfaces. As we regularize surfaces, alternative approaches may consider mesh fairing or denoising strategies on MC surfaces. Most recent techniques consider variational formulations on the vertices positions or the normal vector field to denoise a mesh while preserving its features [32]–[36]. These approaches are extremely efficient to remove noise from a perturbed triangular mesh, but do not generate smooth enough surfaces on low resolution voxel interfaces as illustrated by our experiments (see Sec. 4 and supplementary). A closely related surface regularization approach was proposed in [37]: the fairing variational model is based on squared curvature minimization and imposed hard constraints on the distance to the original vertices. The optimized energy is not convex and this model shares many similarities with the shrouds method. In a similar way, Zhang et al. [38] also propose a Willmore flow to regularize quad meshes as well as hexahedral volumetric ones, with an additional tangential constraint to enhance the quality of the surface elements. However, the non-convexity of the energy and the use of a semi-implicit Euler scheme to minimize it may lead to sub-optimal solutions and numerical instabilities. Our approach relies on a simpler quadratic formulation whose solution has strong geometrical and topological stability properties.

Digital Geometry. To process the geometry or the topology of subsets of $\mathbb{Z}^3$ (*digital objects*), there exist dedicated mathematical tools and efficient algorithms [39]. These approaches exploit either the regularity of the grid, the integer arithmetic or the topology induced by the lattice, to infer some geometrical quantities for instance. Our approach relies on a stable normal vector field estimation on boundaries of the voxel set. To estimate such differential quantities, many strategies exist. We focus here on approaches providing proven multigrid convergence properties such as [40] (see Sec. 3.3 for a more precise definition). From an estimated input normal vector field, we can obtain a piecewise smooth anisotropic normal vector bundle using voting or variational approaches [41], [42]. As we aim to reconstruct a mesh from a set of voxels, we also mention a polyhedrization technique of digital object boundaries [43]. This approach first extracts large patches of *flat* voxels as defined in [44], then apply some iterative fairing steps on the X, Y and Z slices and finally reconstruct an iso-surface. Although authors present interesting results, the overall algorithm is complex with many parameters and heuristic (*e.g.* in the flat patches extraction step). We propose a simpler approach with some asymptotic theoretical properties on the regularized surface.

Contributions. In this article, we propose a variational approach to reconstruct boundaries of voxel sets (Fig. 2) that is fast, robust and easy to implement. Starting from an anisotropic normal vector field estimation on the digital surface, the energy minimization produces a piecewise smooth and fair quadrangulated surface (Fig. 2, second row). The regularized surface is geometrically close to the input digital boundary (Property 1, Sec. 3.3) and guaranteed without self-intersections under easy-to-enforce constraints (Property 2, Sec. 3.4). Since our method preserves the combinatorial structure of the input quadrangulated surface, we can regularize complex multilabel image interfaces and map colorimetric information from the input surface onto

the smooth one for voxel art regularization. Moreover, our variational model is a convex quadratic energy which can be minimized either by solving a sparse linear system, or using an iterative gradient descent scheme which allows topological control and efficient implementation on GPU at interactive frame rates.

In Section 3, we describe the variational model we consider and detail its properties. In Section 4, we experimentally evaluate our tool to regularize digital surfaces, multiple material interfaces and voxel art, and compare it with other methods of the literature. The supplementary material provides detailed proofs of theoretical properties of our model as well as further experimental comparisons with other regularization approaches.

The variational model was first introduced in [15]. Here we extend this result with theoretical stability results of the regularization (Sec. 3.3 and 3.4 and supplementary) and with extensions to multi-labeled images (Sec. 3.5). Furthermore the new GPU optimization can enforce the hypotheses required by the topological stability result (Sec. 3.4). Finally the proposed model is more deeply evaluated and compared with the Shrouds algorithm.

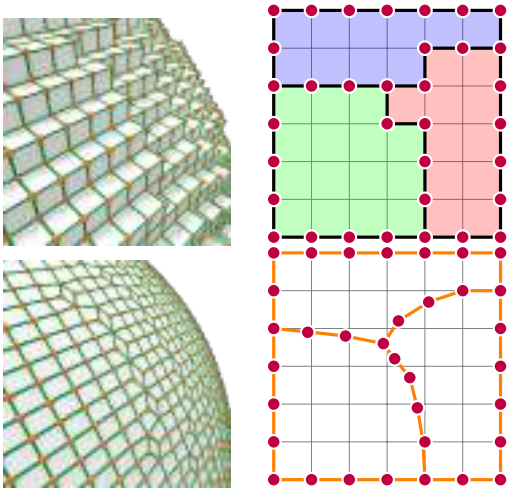


Fig. 2. **Reconstruction of digital surfaces**: given a 3D digital surface (topological boundary of a set of voxels) *(first row left)* and a 2D labeled image with 1D digital interfaces between pixel components *(first row right)*, the proposed approach outputs piecewise smooth reconstructions with the combinatorics as the input surfaces *(second row)*.

3 Variational Regularization

3.1 Digital Surface Model

We first describe the surface model we use to process boundaries of sets of voxels. Let X be a binary voxel object as a subset of $\mathbb{Z}^3$. A convenient way to define the boundary of X considers the cellular inter-pixel decomposition of the grid (see Fig. 2-*left*). The *digital surface* or digital boundary, of X is a collection of *surfels* (cells of dimension 2 embedded as isothetic quads), equipped with some specific adjacency relationship. Digital surface *vertices* are the $0-$cells incident to a surfel. We refer to [39], [45], [46] for topological properties associated to this combinatorial structure. For short, if voxels are connected and adjacent along their faces but not solely along edges or vertices, then the boundary of these voxels form a digital surface that is a closed combinatorial 2-manifold.

Given a multi-labeled image, we can extend this representation to define digital surfaces for all interfaces (Fig. 2-*right*): each connected component of voxels with the same label characterizes a proper digital surface and we can even define an interface between two labels as the connected set of surfels in-between the two voxel sets with these labels.

At any surfel of the digital surface, one can estimate several differential quantities such as a normal vectors. Any algorithm estimating a robust normal vector field on a mesh or a point cloud could be used [41], [47]–[49]. However, estimators dedicated to digital surfaces are usually preferred since they provide multigrid convergence properties [40]: given a smooth object digitized on a grid with grid-step h, the property states that the estimated normal vectors converge uniformly to the normal vectors on the smooth object as h tends to zero. This property is used in Section 3.3 to improve the quality of our reconstruction. For piecewise smooth reconstruction, we can rely on anisotropic normal vector estimation through dedicated approaches [41], [48], [50] or as a post-process of a first normal vector field estimation [42]. Please refer to the supplementary material for experimental comparisons.

In the context of multi-labeled images, normal vectors are estimated on the boundary of each connected component, each surfel having two normal vectors since it is shared by two regions. Averaging their direction (forgetting their orientation) leads to a single normal vector direction per surfel, which is enough input for our variational model.

3.2 Energy formulation

The objective is to regularize the boundaries of digital objects into piecewise smooth ones (Fig. 2). The variational formulation takes as input a combinatorial quad structure with n vertices $P := \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1\dots n}$ and m faces $F := \{f_k\}_{k=1\dots m}$ as quadruples of vertices. We also consider a prescribed normal vector field $\mathbf{n}_f \in \mathbf{S}^2$ for each $f \in F$, whose directions are important but whose orientations can be arbitrary. The output is a quadrangulated surface that has the same combinatorial structure but with vertices $P^* := \{\mathbf{p}_i^*\}_{i=1\dots n}$ such that:

$$P^* := \operatorname{argmin}_{\hat{P}} \ \mathcal{E}(\hat{P}) \tag{1}$$

with

$$\mathcal{E}(\hat{P}) := \alpha \sum_{i=1}^{n} \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2 + \beta \sum_{f\in F} \sum_{\mathbf{d}\,\hat{\mathbf{p}}_j \in \partial f} (\mathbf{d}\,\hat{\mathbf{p}}_j \cdot \mathbf{n}_f)^2 + \gamma \sum_{i=1}^{n} \|\hat{\mathbf{p}}_i - \hat{\mathbf{b}}_i\|^2 . \tag{2}$$

where $\bullet\cdot\bullet$ is the standard $\mathbb{R}^3$ scalar product and $\|\bullet\|$ its associated norm. By a slight abuse of notation, let $\{\mathbf{d}\,\hat{\mathbf{p}}_j\}_{j=1\dots 4}$ be the embedded edges of the face f. Edges do not need to be oriented as we square the scalar product in the second term. See figure 3 for an illustration of each terms. The first term of this quadratic energy is a *data attachment term* to prevent vertices to be too far away from the input digital surface. The second term, the *alignment term*, forces the

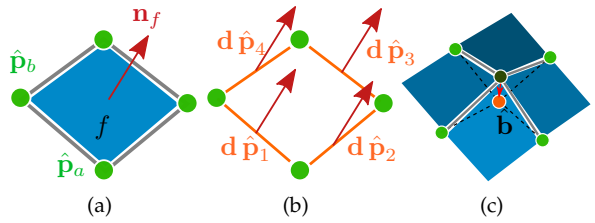


Fig. 3. **Notations** for the normal vector alignment term (a, b) (*e.g.* $\mathbf{d}\,\hat{\mathbf{p}}_1 := \hat{\mathbf{p}}_a - \hat{\mathbf{p}}_b$), and the fairness term (c).

regularized quads to match with the prescribed normal vector. The energy is minimum when all edges belong to the tangent plane defined by $\mathbf{n}_f$. Similar formulation can be found to enforce planarity of quads in many variational formulations [51]. Finally, the last term is a classical *fairness* term which appears in many relaxation geometry processing tasks. If we denote $\hat{\mathbf{b}}_i$ the barycenter of the adjacent vertices to $\hat{\mathbf{p}}_i$, the fairness term forces the points $\hat{\mathbf{p}}_i$ to be close to $\hat{\mathbf{b}}_i$. This term is useful to optimize quad geometry by moving vertices solely along their tangent plane in which the alignment term is already minimal. If $\hat{\mathbf{p}}_i$ belongs to the boundary of $\hat{P}$, no fairness term is considered for this point. In the multi-material case, a specific strategy has been chosen to properly define the barycenter $\hat{\mathbf{b}}_i$, see Sect. 3.5.

Let $\mathbf{p} = \left[\cdots,\ \mathbf{p}_{i-1}^z,\ \mathbf{p}_i^x,\ \mathbf{p}_i^y,\ \mathbf{p}_i^z,\ \mathbf{p}_{i+1}^x, \cdots\right]^T$ be the $3n$ column vector obtained by packing together the coordinates of all input vertices of P. Similarly, we denote by $\hat{\mathbf{p}}$ the $3n$ column vector containing the regularized vertex positions. Energy partial derivatives can be computed from (2) as

$$\frac{\partial \mathcal{E}(\hat{P})}{\partial \hat{\mathbf{p}}_i} := \alpha \sum_{i=1}^{n} 2(\hat{\mathbf{p}}_i - \mathbf{p}_i) + \beta \sum_{f \in F} \sum_{\mathbf{d}\,\hat{\mathbf{p}}_j \in \partial f} 2(\mathbf{d}\,\hat{\mathbf{p}}_j \cdot \mathbf{n}_f)\mathbf{n}_f + \gamma \sum_{i=1}^{n} 2(\hat{\mathbf{b}}_i - \hat{\mathbf{p}}_i) \tag{3}$$

and are linear with respect to $\hat{\mathbf{p}}$ and $\mathbf{p}$. If we represent $\frac{\partial \mathcal{E}(\hat{P})}{\partial \hat{\mathbf{p}}_i}$ as a column vector $\nabla\mathcal{E}(\hat{P})$, there exists two discrete linear maps ($3n \times 3n$ matrices) $\mathbf{A}$ and $\mathbf{B}$ such that $\nabla\mathcal{E}(\hat{P}) = \mathbf{A}\hat{\mathbf{p}} - \mathbf{B}\mathbf{p}$. Minimizing (2) can be done by solving the linear system $\nabla\mathcal{E}(P^*) = \mathbf{0} \Leftrightarrow \mathbf{A}\mathbf{p}^* = \mathbf{B}\mathbf{p}$. Note that $\mathbf{A}$ is sparse, which allows an efficient resolution of the linear system (see supplementary material). For some specific applications such as interactive regularization on GPU, a gradient descent using (3) may be preferred.

Positive weights α, β and γ can be set independently to emphasize some of the terms. As illustrated in the experiments (Sect. 4 and Fig. 15), each term as an explicit impact on the expected result: its proximity to the original surface, its conforming to the given normal vector field and the quality of the quadrangulation respectively. As long as $\alpha > 0$, the operator is full rank, and the energy admits a unique solution. In our experiments except the ablation study, we considered the same parameters for all figures: $\alpha = 10^{-3}$, $\beta = 1$ and $\gamma = 10^{-1}$; The rationale being that the alignment term is the most important one, followed one order of magnitude lower by the fairness term to allow additional tangential regularization. The attachment term gets the lowest weight since it is mainly present to ensure that the variational problem has a unique solution: the other terms have indeed non null kernel.

3.3 Proximity and stability

An important property of the proposed formulation is that we can guarantee a proximity between the regularized surface and the input digital object boundary. A more formal analysis considers an asymptotic multigrid behavior of the surface: given a smooth shape $\mathcal{X} \subset \mathbb{R}^3$ and its digitization X_h on a grid step h, *e.g.* using the Gauss digitization $X_h := \mathcal{X} \cap (h \cdot \mathbb{Z}^3)$. We would like to know how far is the regularized quadrangulation of ∂X_h with respect to $\partial\mathcal{X}$ as h tends to zero.

We have the following proximity and stability result, which assumes that the fairness term is negligible compared to the others.

Proposition 1 (Proximity and stability)**.** *Let P be the input voxel vertices of the boundary of X_h, and let P^* be the output vertices of our regularized surface. Then the average distance between P^* and P is upper-bounded by $O(h)$. Furthermore, the same holds between P^* and $\partial\mathcal{X}$. More precisely,*

$$\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{p}_i^* - \mathbf{p}_i\| \leq C \cdot h\,, \quad \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{p}_i^*, \partial\mathcal{X}) \leq C' \cdot h\,, \tag{4}$$

for some constants C, C' that depend uniquely on the reach (say greater than R) of $\partial\mathcal{X}$, and for a gridstep h, $0 < h < 2R/\sqrt{3}$.

Proof. Sketch of the proof (full proof in the supplementary material). The main idea is to exhibit a quadrangulated surface $\tilde{P}$ built on the continuous boundary $\partial\mathcal{X}$, whose energy $\mathcal{E}(\tilde{P})$ is upper bounded by a constant. The energy $\mathcal{E}(P^*)$ of the optimal shape is then no greater than this constant. The relations above are then induced by this inequality. More precisely, the proof follows these steps.
1. Since X_h is the digitization of a solid $\mathcal{X}$ whose boundary has positive reach, then $\partial\mathcal{X}$ is $\sqrt{3}h/2$-Hausdorff close to ∂X_h, for small enough gridstep h ([52], Theorem 1). Then the orthogonal projection ξ onto $\partial\mathcal{X}$ is well defined for ∂X_h, being within the reach. Every projected vertex $\tilde{\mathbf{p}}_i := \xi(\mathbf{p}_i)$ onto $\partial\mathcal{X}$ is thus at distance no greater than $\sqrt{3}h/2$ to $\mathbf{p}_i$.
2. Denoting $\tilde{P}$ the set of projected vertices $\tilde{\mathbf{p}}_i$, we evaluate an upper bound for its energy $\mathcal{E}(\tilde{P})$ (its edge/face combinatorics is the same as the input surface). The three terms of (2) can be upperbounded as (n, m correspond to the number of vertices and faces resp.):

$$\mathcal{E}(\tilde{P}) \leq \alpha\frac{3}{4}nh^2 + \beta K m h^{2+2\delta} + \gamma(1+\sqrt{3})^2 nh^2.$$

The first and third terms follow easily from $\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|_\infty \leq \sqrt{3}h/2$. Middle term requires the fact that the normal onto a smooth surface does not vary too quickly. Last, symbol $\delta \geq 0$ indicates the speed of convergence of the prescribed normal vector field $\mathbf{n}_f$ toward the true normal of the surface $\partial\mathcal{X}$ at this point.
3. From ([52], Lemma 10), the area of ∂X_h is upper bounded by a constant times the area of $\partial\mathcal{X}$, thus mh^2 is upper bounded. Since $n = m + cste$ on closed digitized surfaces, we have $n = O(1/h^2)$ and $m = O(1/h^2)$. So $\mathcal{E}(\tilde{P})$ is upper bounded by a constant, even for non-convergent prescribed normal vector field ($\delta = 0$) !

4. P^* being optimal, we have $\mathcal{E}(P^*) \leq \mathcal{E}(\tilde{P}) \leq K'$, for K' a constant. Since all three terms of the energy are positive, keeping the first term and Cauchy-Schwartz inequality induce $\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{p}_i - \mathbf{p}_i^*\| = \sqrt{\frac{K'}{\alpha}}\frac{1}{\sqrt{n}}$. Last, the Jacobian of ξ being bounded and $\xi(\partial X_h)$ covering $\partial\mathcal{X}$, we get $n = \Omega(1/h^2)$ which concludes (4), left. Equation (4), right, is immediate from the former and $\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\| \leq O(h)$. □

The proposition is stated for digital surface P defined as the boundary of the Gauss digitization of some smooth manifold $\mathcal{X}$, using a preliminary step resulting from [52] that bounds the Hausdorff distance between P and $\partial\mathcal{X}$ by $O(h)$. When dealing with smooth surface with boundaries, the Gauss digitization framework may not make sense but Proposition 1 would also hold as long as we can define P as some digitization of the surface at Hausdorff distance $O(h)$.

A remarkable aspect of this result is that it does not depend on the quality of the input normal vector field. It means that the regularization is very stable in terms of vertex position, while aligning edges so that most of them are orthogonal to the input normal vector field.

As also shown in the supplementary material, the quality of the input normal field impacts the normal field of the output surface, since we show these two fields are close and gets closer as h decreases. If we use normal vector field estimators that converge toward the normal vector bundle of $\partial\mathcal{X}$ (for instance, the digital integral invariant estimator has a pointwise estimation error upper-bounded by $O(h^{2/3})$ [40], [53]), then the regularized surface has a normal field that is close to the normal field of ∂X while being close to ∂X on average. In other words, it is almost the same surface. We provide additional experimental results to illustrate the robustness of the reconstruction with respect to the input normal vector estimator or to noise in the estimation (see supplemental material).

3.4 Topological guarantees

When minimizing (2) on noisy input or thin structures, self-intersections may appear in the reconstruction. We describe here a simple technique to enforce topological guarantees on the reconstructed surface. Let us consider a generic embedding into $\mathbb{R}^3$ of the reconstructed polygonal face f_k of $\hat{P}$. In our experiments, we illustrate the reconstructed surface $\hat{P}$ using a piecewise linear embedding of quads (using an arbitrary triangulation). Different or higher order embeddings could also be considered and our results may also apply in these cases.

We thus consider the following sufficient conditions to prevent self-intersections in $\hat{P}$ (with h the grid step):

(i) Each point in the embedding of f_k must lie in the convex hull of its vertices.

(ii) Each vertex $\hat{\mathbf{p}}_i$ of $\hat{P}$ must stay in the closed cube of edge $h - \epsilon$ centered on its corresponding vertex $\mathbf{p}_i$ of P, for an arbitrary small $\epsilon, 0 < \epsilon < h$.

In Appendix, we prove that if both conditions are satisfied, then no self-intersections can occur in $\hat{P}$. More precisely we have:

Proposition 2 (Topological guarantees)**.** *If rules (i) and (ii) are satisfied for $\hat{P}$, then the optimal mesh P^* has no self-intersections. More formally, two faces of P^* either have no intersection or, when they intersect it is along their common edge segment or at their common vertex.*

Condition (i) is quite easy to fulfill as many natural embeddings of faces lie in the convex hull of its vertices: e.g. two triangle decomposition (used in illustrations), barycentric subdivision, or even a bilinear interpolation of vertex positions.

Concerning condition (ii), forcing vertices to stay in a fixed neighborhood would imply additional linear constraints in the optimization problem (1). Even if more elaborate methods could have been used (e.g. subspace minimization [37]), we have considered in Fig. 4 a simpler approach using a gradient descent minimization of (1) with a clamping of the vertices during the advection step (by setting to zero some components of the gradient vector according to the local configuration). As illustrated in Fig. 4, the clamping step has an impact on the smoothness of the reconstruction. We recall that (i) and (ii) are sufficient conditions to prevent self-intersection. In Sec. 4, the clamping has not been used as no self-intersection has been observed.

Since the combinatorial structures of $\hat{P}$ and P are identical, Prop. 2 implies that both surfaces share the same topological properties. For instance, if P is globally a 2-manifold surface (*e.g.* boundaries in well-composed pictures [45]) or if it is locally a 2-manifold (*e.g.* when coming from the digitization of continuous objects with specific regularity properties [52]), then the regularized $\hat{P}$ has the same property.

3.5 Multi-material

Even if the variational formulation (2) is a quadratic global energy, its evaluation and its gradient are local and can be extended to reconstruct complex multi-labeled interfaces. On multi-labeled images, we rely on the limited number of topological configurations of digital interfaces (Fig. 5): each surfel is shared by two voxels with different labels and each edge is a junction of at most 4 interfaces. Let us assume that we have an estimated normal vector per surfel (see Section 3.1).

The first term (data attachment) in (2) is purely local on vertices and is perfectly adapted to the multi-labeled interfaces. Thanks to the *surfel-to-edges* formulation of the second term (alignment term, see Fig. 3-(a, b) and 5-(b)), each junction edge contributes more than twice to the energy but the overall formulation does not change. Therefore the vertices of such edges take into account contributions of all adjacent normal vectors. For the third term optimizing the quad geometry, each barycenter considers all adjacent vertices, possibly on different interfaces. Another option would have been to simply cancel out this term for vertices on junction edges but regularizing their position with respect to their neighbors allows us to regularize 1-D junctions between interfaces (see Fig 11, first column).

As a consequence, the energy model and its minimization described in Section 3 can also be applied to multi-labeled interfaces. Whatever the topology of the input combinatorial structure, the linear operator representing the gradient of (2) is still sparse and invertible.

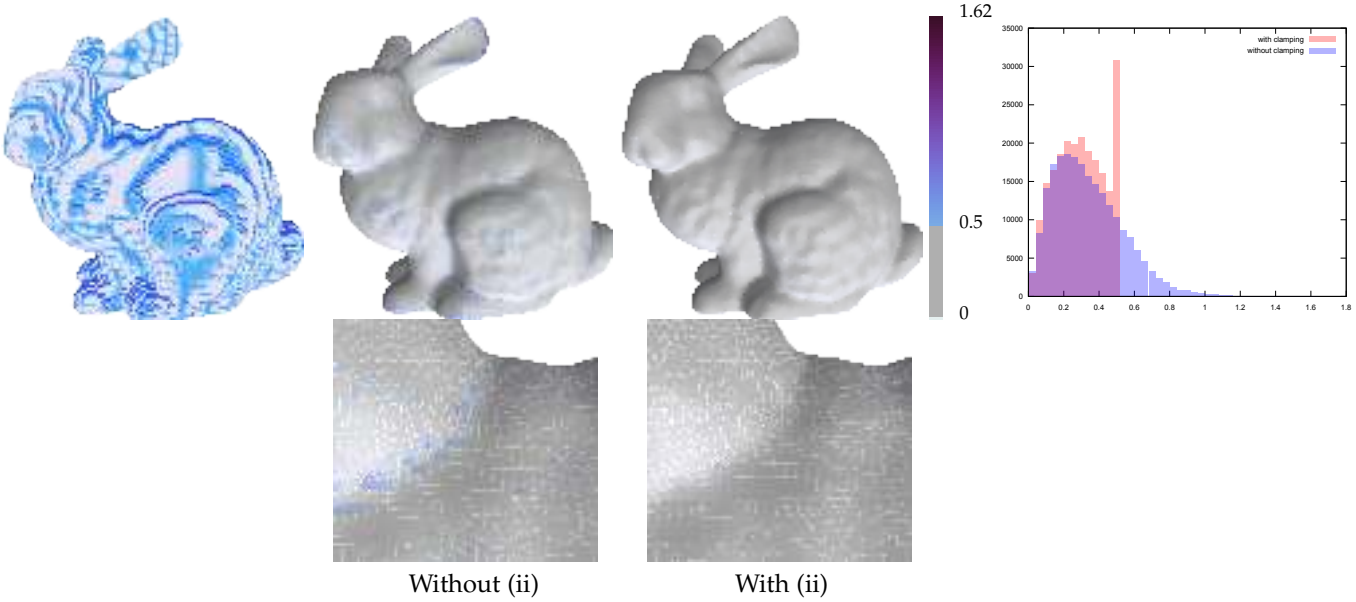


Fig. 4. **Topological control:** given a 128^3 `Stanford bunny` shape $(left)$, we colormap the distances $\|\mathbf{p}_i^* - \mathbf{p}_i\|_\infty$ for the reconstruction minimizing (2) $(middle, right)$, and for the reconstruction from (2) when considering the clamping step during the vertices advection $(middle, right)$. This step may alter the smoothness of the reconstruction, but ensures that condition (ii) of Prop. 2 is satisfied.

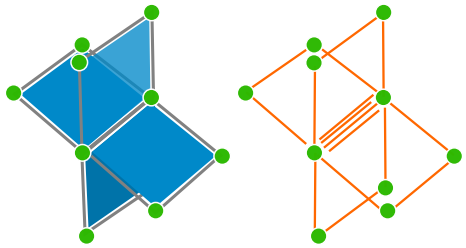


Fig. 5. **Handling multiple interfaces:** junction between interfaces in the digital grid (worst case with four different labels) in (a) and the decomposition into independent faces/edges to evaluate the energy function (b).

4 Experiments

4.1 Implementation details

Two different implementations are available. The first one is a direct minimization of (2) using a sparse matrix linear solve on the gradient operator, following [15]. This approach requires to explicitly compute the operator from the digital surface structure (sparse $3n \times 3n$ matrix, see Supplementary), and then use a classical linear algebra solver (*e.g.* from [54]). This implementation is publicly available in the DGtal library (release 1.1) [55], [56].

Alternatively, for interactive purposes, we can update the vertex positions on GPU using a simple gradient descent. Using only a few steps, we quickly obtain a close approximation of the optimal solution found by solving the linear system. Timings are given in Section 4.4. In all following experiments, when not specified, we have used the anisotropic normal vector estimation described in [42].

For the experimental evaluation, we have considered our own implementation of the shrouds approach [27] that may differ slightly from the original version as the authors code is not available. Implementation details are given in supplementary material and our code is publicly available in the DGtal library [55], release 1.1 [56].

In the experiments of Figures 4 and 15, the topological clamping constraint (ii) has been considered. For all the other tests, we have not considered such constraint as no topological issues have been observed in the regularized surface.

4.2 Mono-label surface reconstruction

We first compare our regularization to alternative approaches on the boundary of a single set of voxels. We mainly compare our method to two classical iso-surface contouring techniques: the Marching-Cubes (MC) [16] and the Dual-Contouring (DC) [23] (see Fig. 6). On binary voxel sets, the MC approach generates an iso-surface between the voxel set and its complement using local patches of triangles. DC approaches construct an iso-surface from Hermite data (position and normal vectors) edges of the dual grid (see Fig. 6). Given position and normal vectors of the implicit function on the lattice grid, DC generates vertices within local configurations of voxels minimizing a quadratic error function (QEF): the vertex $\mathbf{v}$ in each configuration minimizes

$$\sum_j (\mathbf{n}_j \cdot (\mathbf{v} - \mathbf{s}_j))^2 \,,$$

where $\{\mathbf{s}_j\}$ and $\{\mathbf{n}_j\}$ denotes the positions and normal vectors on the edges of the voxel configuration (see Fig. 6). As detailed in the supplementary material, the quality of the DC surface is very sensitive to the quality of the input positions and normal vectors.

In Figure 7, we compare several approaches on an object with different voxel sizes. In this experiment, DC input normal vectors are obtained as gradients of the binary map (please refer to the supplementary material for further discussions). In our approach, we have considered both trivial normal vectors (similarly to DC) and quad perpendicular

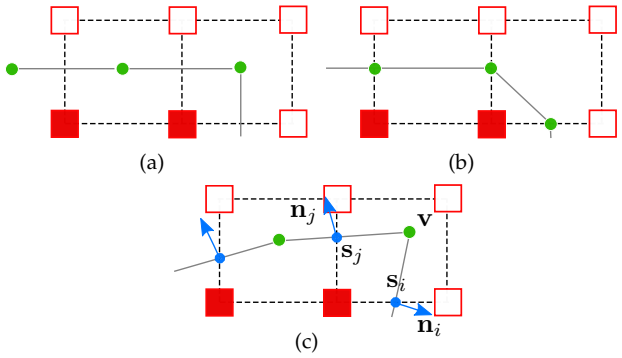


Fig. 6. **Classical contouring techniques,** representing P on the dual grid where squares (empty or not) represent the voxels (inside or outside) (a): the MC sets vertices in-between two voxels with different labels (b) whereas DC sets vertices within voxel configurations from Hermite data (position and normal vectors in blue) (c).

unit vectors and normal vectors obtained from [42]. For a complete comparison, we have also considered a state-of-the-art mesh denoising approach on the MC surface using [36], and the shrouds approach [27].

As illustrated in Figure 7, our regularization approach produces smoother and more regular surfaces, even on very low resolution voxel sets with very poor quality input normal vectors *(fifth column)*. When using anisotropic normal vector field *(sixth column)*, we better capture the surface singularities. Note that the same α, β and γ parameters have been used for all shapes. Furthermore, we keep a mapping between the original and regularized quads, which will be used for voxel art regularization later in Section 4.3. Results on higher voxel resolution are given in Figure 8. In the supplementary material, we explicit the theoretical proximity and stability results of our approach.

In Figure 9, we highlight the impact of the fairness term in the regularization: the aspect ratios of quads (shortest diagonal by the longest one) are optimized when using a fairness term leading to a more regular surface.

4.3 Multi-labeled surfaces and voxel art reconstruction

In Figure 10, we first consider a simple multi-labeled image defined by four shapes (cube, torus, sphere, cone). Following Section 3.5, our approach can regularize all interfaces in a way that each patch is a combinatorial 2-manifold with perfect sewing on non-manifold edges (edges adjacent to at least 3 different labels). Thanks to the stability of the input normal vector field and the smoothness of the regularized surface, no self-intersection of the interfaces have been observed, even on complex shapes. In Figure 10, we have also experimented the volumetric tetrahedrization approach of CGAL [57] (using both a Lloyd's optimization and a sliver exudation process). Note that CGAL allows the user to explicitly define additional 1D features (polylines) the volumetric reconstruction has to preserve. This option has not been considered in this test as it would have only impacted the 1D boundaries between interfaces and not the regularity of the overall surface mesh (less smooth on the low resolution voxel input). Furthermore, extracting 1D smooth feature polylines is a problem that needs to be solved first (defining the feature lines as sequences of edges of the original digital surface would have produced similar results on low resolution domain).

In Figures 1 and 11, we illustrate our final result on low-resolution voxel arts: we use the one-to-one mapping between input voxel quads and regularized ones to transfer colorimetric or material information to the smooth reconstruction. The first row of Figure 11 illustrates the color mapping on a single digital surface. Note that even on thin voxel structures, our approach is able to reconstruct a smooth and self-intersection free surface. From the input labeled voxels, instead of creating interfaces per label, we can decide which set of labels defines a *region* that is later used to define the interfaces. For instance, in the second column of Figure 11, we have defined two regions, one for the helmet glass and one for the rest of the object. The mapping between quads allows us to project voxel labels (and thus colors) onto the regularized surfaces. In the third column, four regions have been defined: one for the sea, one for the ship, one for the cloud and one for the islands. One can see that the regularized surfaces perfectly reflects the voxel art aspect with piecewise smooth appearance.

4.4 Performances

In Table 1, we provide the performances of our regularization approach. Following a normal vector field estimation given by [53], we detail timing of the energy minimization using a CPU linear solver ([54], `SimplicialLDLT` solver for sparse matrix on single-core Intel Xeon E5); and a GPU based iterative conjugate gradient solver (OpenCL implementation on a AMD FirePro D500). The iterative GPU solver can also be used to visualize the surface interactively during regularization. As illustrated in supplementary video, less than 20 steps are enough to obtain a visually good surface for the `dragon` object (≈11ms per step).

4.5 Extensions

A classical drawback when performing some geometry processing tasks on digital surfaces is that the original isothetic quad surface may not capture the proper metric of the underlying smooth object. As illustrated in Figure 12 (a), computing exact geodesics, using [58], on the original surface exhibits some non-isotropic distance artefacts. Thanks to the stability results (Sec. 3.3), we can perform the computation on the regularized surface and map back the results onto the digital object (Fig. 12 (b)). Note that this geometrical proxy can be either explicitly represented, or implicitly computed when performing the task.

As our reconstruction framework performs well on low resolution voxel geometry, we can upscale coarse voxel objects by reconstructing their interface and voxelizing them to higher resolution grids (cf. Fig. 13). Note that in this process, one can voxelize directly the regularized quad surface without the need of explicitly representing it.

Finally, additional constraints could be considered in the energy formulation of (2). For instance, we illustrate a simple update of the fairness term, which takes into account colorimetric information of labeled voxels, in order to obtain a texture aware quadrangulation. On the ball example of Fig. 11 *(first row, second column)*, the fairness term regularizes the geometry of the quad with tangential constraints which

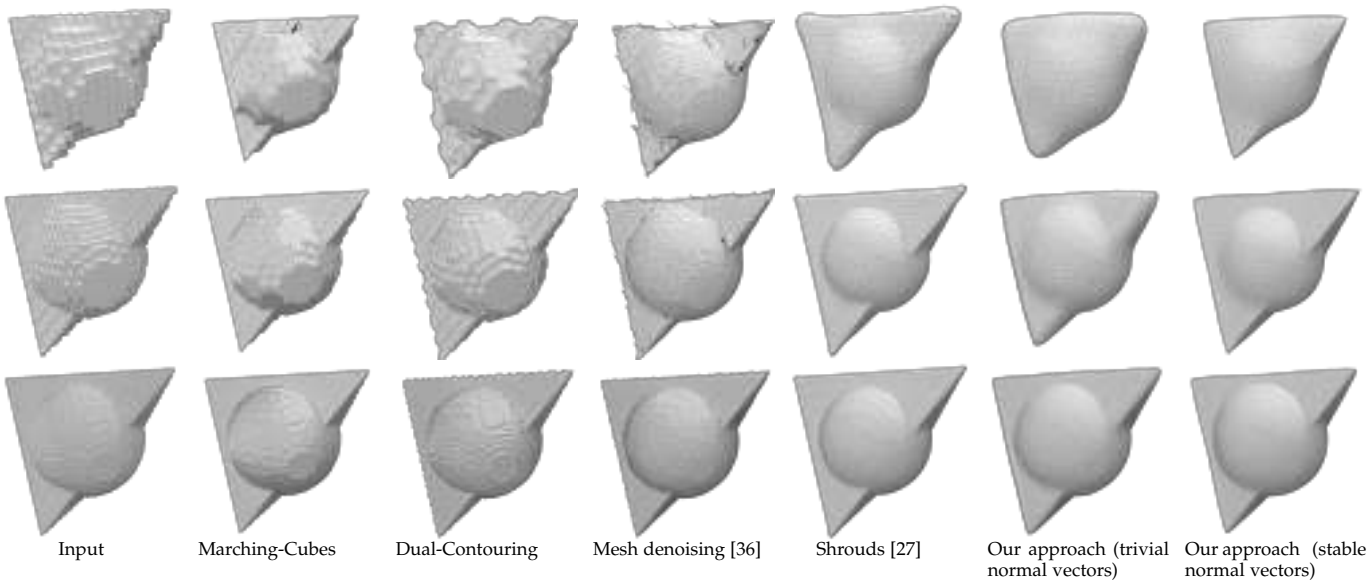


Fig. 7. **Comparisons with existing approaches on low resolution voxel objects** (`ball+tet` shape, 10^3, 20^3 and 40^3 bounding boxes). For the last two columns (our results), we have used the same weights for all shapes ($\alpha = 10^{-3}$, $\beta = 1$ and $\gamma = 10^{-1}$).

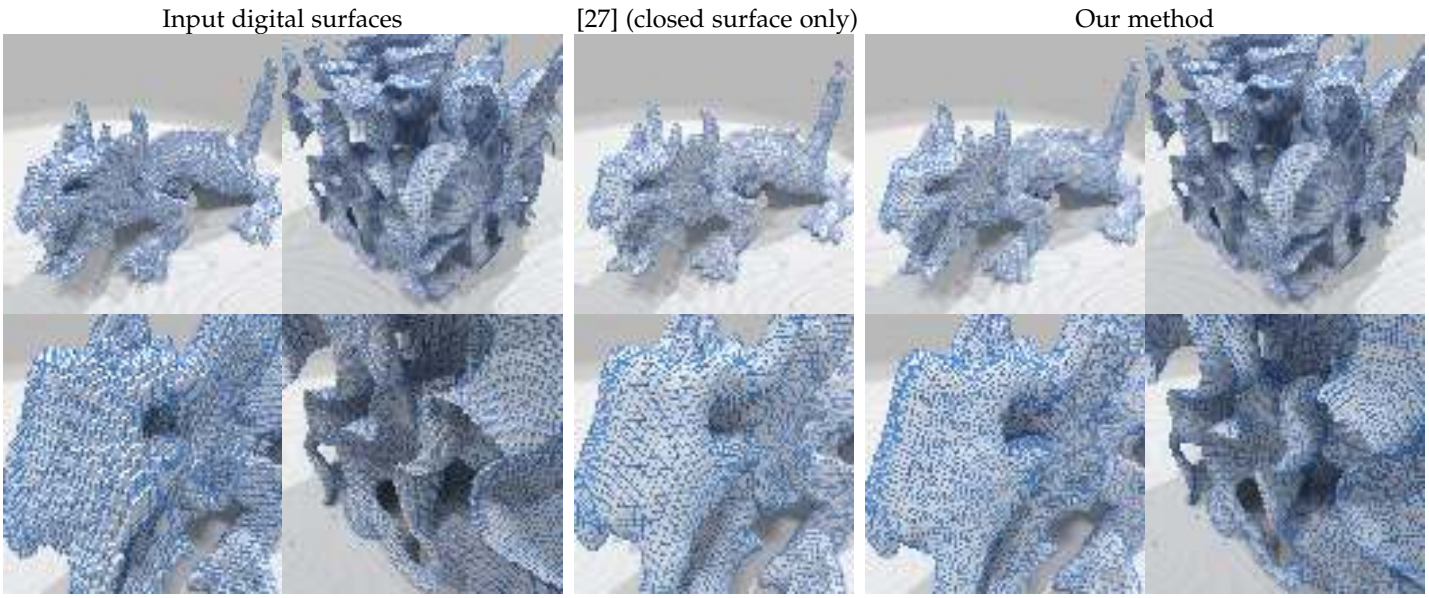


Fig. 8. **Regularization on high resolution voxel shapes:** Our reconstruction technique allows us to generate high quality quadrangulation for high resolution objects(256^3 for the `Dragon` and 128^3 for the granular material). The Shrouds approach [27] also produces a smooth output but only for closed surfaces, and with many degenerate faces (see Fig. 9).

TABLE 1
Performances. The normal vector fields have been extracted using [53] (with radius $r = 4$). The precomputation step consists in preparing the linear operator encoding the regularization problem. For the iterative solver on GPU, timings have been obtained with the following stopping criterion: $\|\nabla \mathcal{E}\|_2 < 10^{-3}$ (timings are given in seconds)

Object	Domain	# of quads	Normal vectors	Precomputation	CPU Solve	GPU Solve
`ball+tet`	40^3	20838	2.280	1.052	13.379	0.494 (689.183 iter/s)
`bunny`	128^3	55262	5.559	2.317	49.831	1.264 (206.428 iter/s)
`dragon`	256^3	104916	10.773	4.282	52.103	3.600 (105.812 iter/s)
`granular`	256^3	236712	32.532	11.900	382.202	6.644 (48.3137 iter/s)
`torus`	256^3	367660	68.265	23.000	1001.160	1.695 (35.9826 iter/s)

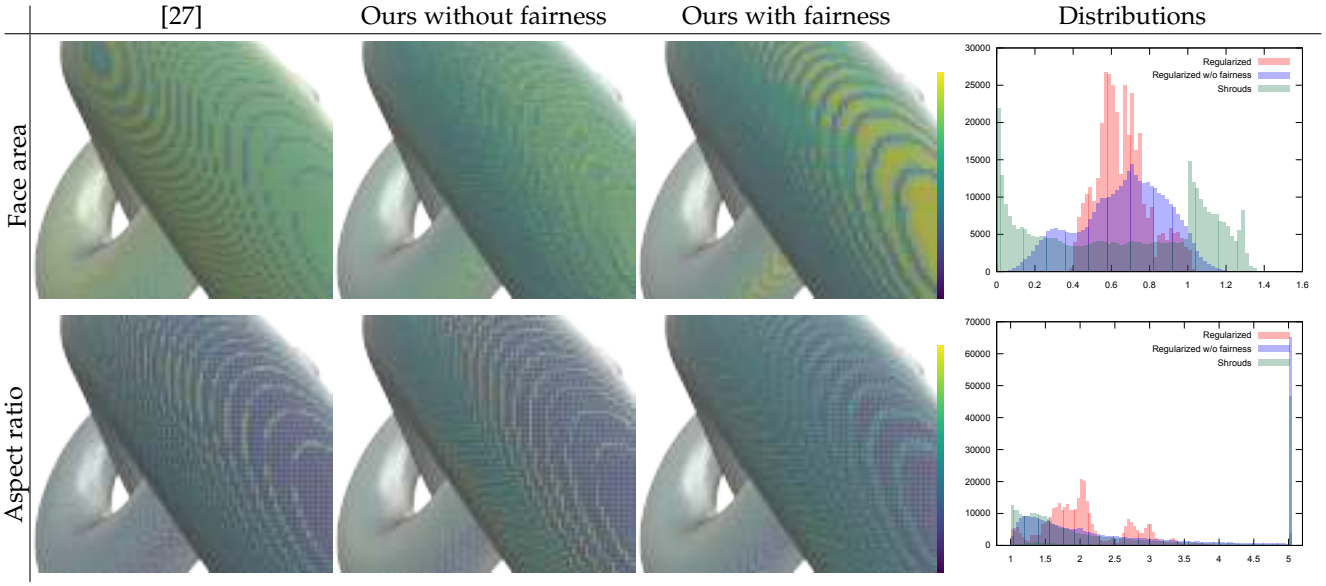
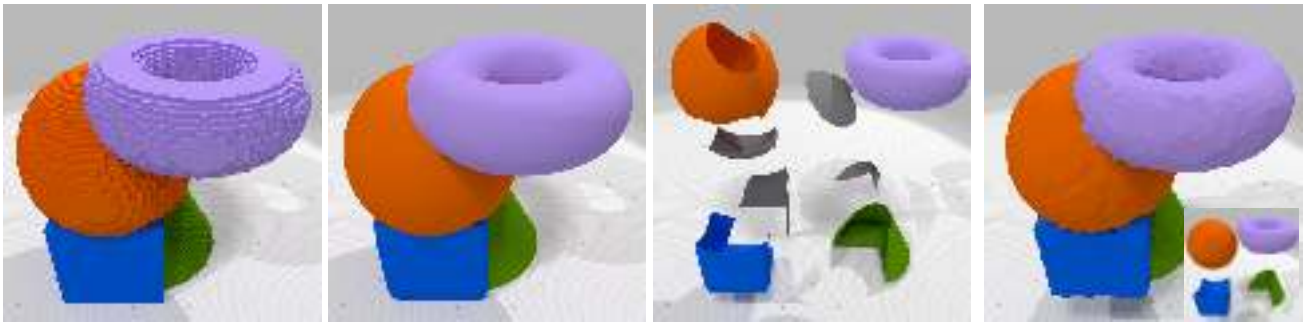


Fig. 9. **Quality evaluation of the polygonal faces:** When considering the surface area or the aspect ratio (ratio between the third and the second eigenvalue of the vertices covariance matrix) of the faces (polygonal for [27], quads in our case), we first observe that [27] produces many degenerates faces (aspect ratio for this approach has been clamped to 5 in the second row). Even without the fairness term in the energy formulation, our regularization approach produces more uniform faces, which is further improved using the fairness term.



(a) (b) (c) (d)

Fig. 10. **Multi-labeled image example**: given the input shapes of (a) in a 75^3 domain, (b) and (c) illustrate the regularization obtained by our approach ($\alpha = 10^{-3}$, $\beta = 1$ and $\gamma = 10^{-1}$). In (d) we have the interfaces obtained using a volumetric tetrahedrization from [57].

makes zigzags along the texture discontinuities. A simple workaround consists in discarding some neighbors in the barycenter computation of the fairness term according to the local configuration of neighboring labels (Fig. 14). This will force some quad edge alignments without changing the global convexity of Eq. (2). We leave more complex intrinsic constraints, *e.g.* alignments of quads with respect to a prescribed crossfield, as interesting future works.

5 Discussions and Limitations

Our reconstruction approach efficiently produces piecewise smooth and regular quadrangulation from labeled images of low to high resolutions. Thanks to the mapping between digital interfaces quads to regularized ones, we can transfer information from the original data (*e.g.* material information) to the smooth version. Our approach relies on an input normal vector field estimation that is used in a quadratic variational formulation whose minimization produces the results. As illustrated in the ablation study in Fig. 15, each term of the energy function has a controllable impact in the final result.

The main limitation concerns the interactive interaction with the voxel objects. Our minimization can be implemented efficiently on GPU producing real time or interactive-time regularization. The main bottleneck is the input normal vector field estimated on CPU at this point. An interesting future work would consist in estimating the normal vectors on GPU and thus in being able to visualize smooth surfaces while interactively modifying the voxel geometry. In such voxel based geometric modeling context, larger scenes would require adaptive or hierarchical approaches. Our energy formulation could be extended to handle irregular quad inputs (*e.g.* boundary of multiresolution voxel sets), but the main challenge would be to design efficient data structures to perform fast updates of the energy gradient.

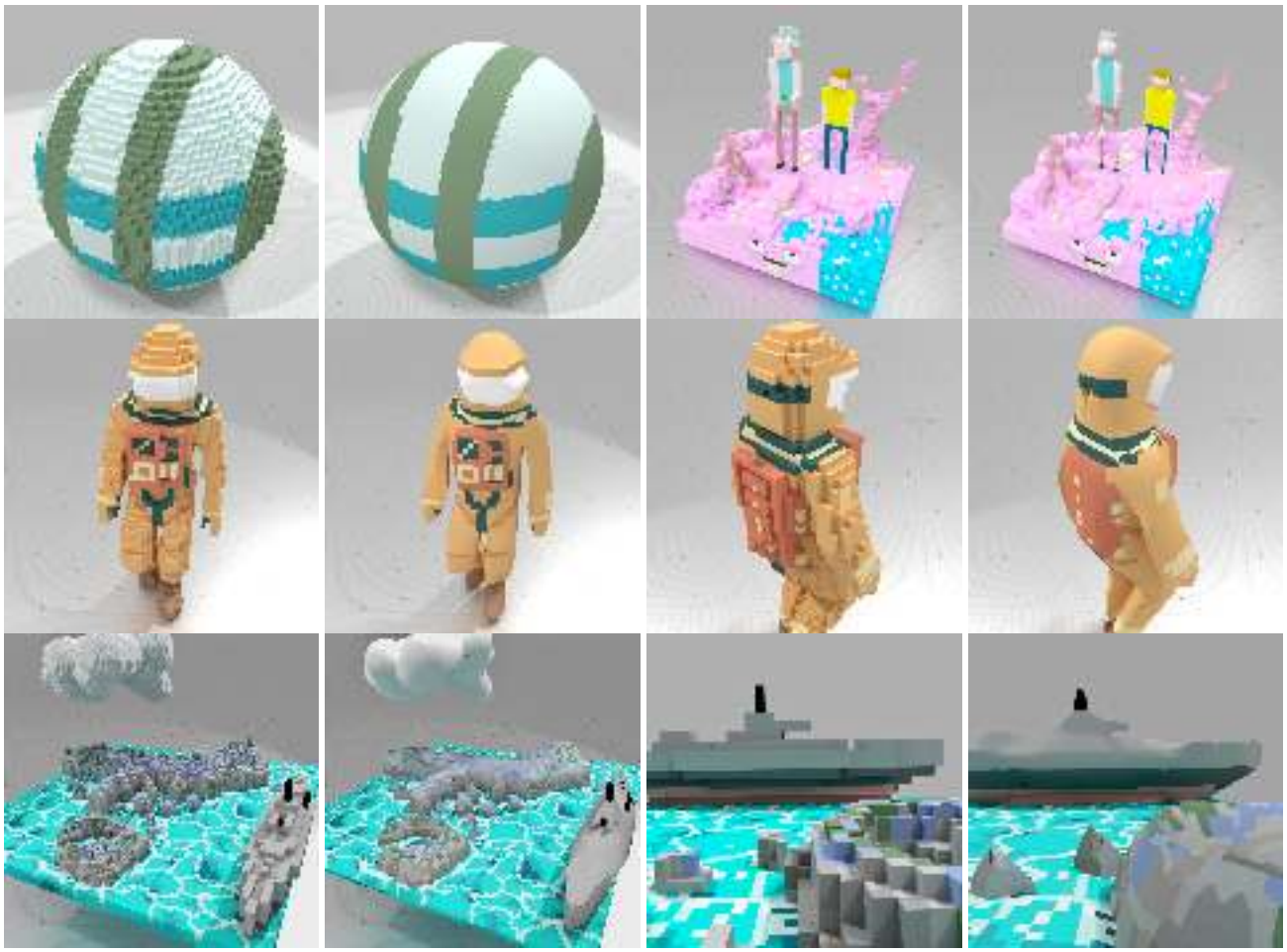
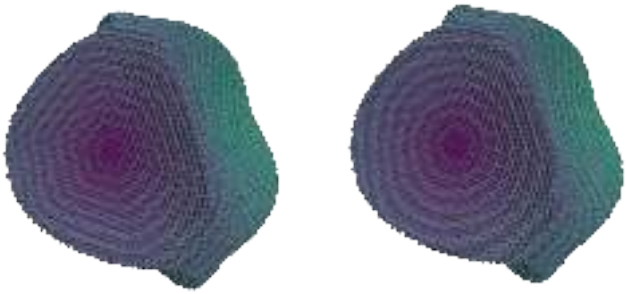


Fig. 11. **Voxel art reconstruction results**. First row, a 50^3 ball and a 45^3 voxel scene with very thin structures. Second row: $25 \times 20 \times 65$ volume with 2 regions, one for the helmet glass and one for the rest. Third row: 127^3 volume with four regions: the sea, the ship, the cloud and the island. All experiments have been obtained with the same parameters ($\alpha = 10^{-3}$, $\beta = 1$ and $\gamma = 10^{-1}$). Voxel artwork courtesy of Elbriga and Mike Judge.



(a) (b)

Fig. 12. **Regularization as a geometrical proxy**: When performing some geometry processing on the digital surface (*e.g.* exact geodesics), the anisotropic nature of digital surfaces may induce metric issues (a). When using the smooth regularized surface as a proxy on which the computation is implicitly performed, we can use the one-to-one mapping of quads to project back the results onto the original digital surface (b).

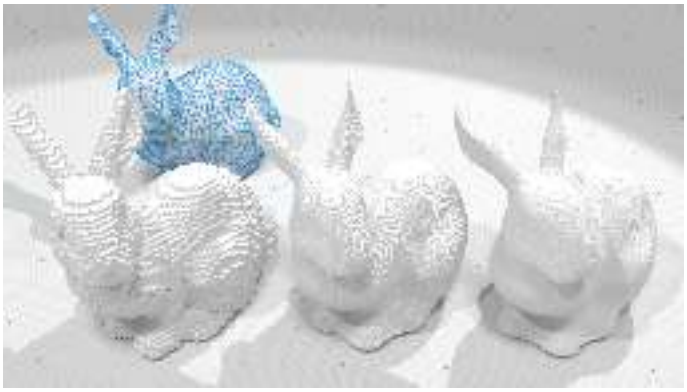


Fig. 13. **Voxel upscaling**: starting from a low resolution `Bunny` object (64^3), we can reconstruct its surface (blue piecewise smooth surface) and use it to upscale the input object to 256^3 and 512^3 resolutions (middle and right).

ACKNOWLEDGMENT

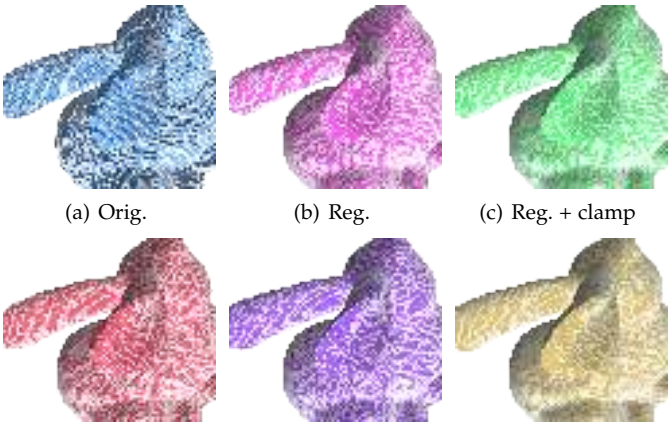
This work has been partly funded by the CoMeDiC ANR-15-CE40-0006 research grant.

REFERENCES

[1] V. Kämpe, E. Sintorn, and U. Assarsson, "High resolution sparse voxel dags," *ACM Transactions on Graphics*, vol. 32, no. 4, Jul. 7, 2013, SIGGRAPH 2013.

(a) (b) (c)

Fig. 14. **Local Fairness term adjustment:** on labeled data, we can locally adjust the fairness term in order to force the quad edges to align with the label discontinuities. For instance, we can discard some neighbors when computing the barycenters $\left\{\hat{\mathbf{b}}_i\right\}$ (a) according to the configuration of adjacent face labels, leading to label specific reconstruction $(b-c)$.



(a) Orig. (b) Reg. (c) Reg. + clamp

(d) $\beta = 0$ (e) $\gamma = 0$ (f) $\gamma = 0$ + clamp

Fig. 15. **Ablation study:** 128^3 `Stanford bunny` shape (a) and its regularization with $\alpha = 10^{-3}, \beta = 1.0, \gamma = 10^{-1}$ (b). We compare the results with the gradient clamping (c), without the alignment term (d), without the fairness term (e) and without fairness nor clamping (f). Note that setting the alignment and fairness terms to zero corresponds to (a).

[2] A. J. Villanueva, F. Marton, and E. Gobbetti, "Ssvdags: Symmetry-aware sparse voxel dags," in *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '16, Redmond, Washington: ACM, 2016, pp. 7–14, ISBN: 978-1-4503-4043-4.

[3] A. J. Villanueva, F. Marton, and E. Gobetti, "Symmetry-aware sparse voxel DAGs (SSVDAGs) for compression-domain tracing of high-resolution geometric scenes," *Journal of Computer Graphics Techniques (JCGT)*, vol. 6, no. 2, pp. 1–30, May 2017, ISSN: 2331-7418.

[4] C. Wojtan, M. Carlson, P. J. Mucha, and G. Turk, "Animating corrosion and erosion.," in *NPH*, 2007, pp. 15–22.

[5] M. D. Jones, M. Farley, J. Butler, and M. Beardall, "Directable weathering of concave rock using curvature estimation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 1, pp. 81–94, 2010.

[6] N. Maréchal, E. Guérin, E. Galin, S. Mérillou, and N. Mérillou, "Heat transfer simulation for modeling realistic winter sceneries," in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 449–458.

[7] C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann, "Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering," in *SI3D*, ACM, 2009, pp. 15–22.

[8] S. Thiedemann, N. Henrich, T. Grosch, and S. Müller, "Voxel-based global illumination," *Symposium on Interactive 3D Graphics and Games on - I3D '11*, p. 103, 2011. DOI: 10.1145/1944745.1944763.

[9] K. Viktor, E. Sintorn, and U. Assarsson, "Fast , Memory-Efficient Construction of Voxelized Shadows," *I3D*, vol. 22, no. 10, p. 6, 2015.

[10] @ephtracy, *Magicavoxel*. [Online]. Available: https://ephtracy.github.io.

[11] J. Kopf and D. Lischinski, "Depixelizing pixel art," in *ACM Transactions on graphics (TOG)*, ACM, vol. 30, 2011, p. 99.

[12] T. Hildebrand, A. Laib, R. Müller, J. Dequeker, and P. Rüegsegger, "Direct three-dimensional morphometric analysis of human cancellous bone: microstructural data from spine, femur, iliac crest, and calcaneus.," *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research*, vol. 14, no. 7, pp. 1167–74, Jul. 1999, ISSN: 0884-0431.

[13] F. Flin, J. B. Brzoska, B. Lesaffre, C. Coléou, P. Lamboley, D. Coeurjolly, O. Teytaud, G. Vignoles, and J. F. Delesse, "An adaptive filtering method to evaluate normal vectors and surface areas of 3d objects. application to snow images from x-ray tomography," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 585–596, 2005.

[14] J. Dardenne, S. Valette, N. Siauve, N. Burais, and R. Prost, "Variational tetrahedral mesh generation from discrete volume data," *The Visual Computer*, vol. 25, no. 5-7, pp. 401–410, 2009.

[15] D. Coeurjolly, P. Gueth, and J.-O. Lachaud, "Digital surface regularization by normal vector field alignment," in *20th International Conference on Discrete Geometry for Computer Imagery*, ser. 20th International Conference on Discrete Geometry for Computer Imagery, vol. LNCS, Vienna, Austria, 2017.

[16] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, ACM, vol. 21, 1987, pp. 163–169.

[17] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.

[18] J. Lachaud and A. Montanvert, "Continuous analogs of digital boundaries: A topological approach to iso-surfaces," *Graphical Models*, vol. 62, no. 3, pp. 129–164, 2000.

[19] Z. Wu and J. M. Sullivan, "Multiple material marching cubes algorithm," *International Journal for Numerical Methods in Engineering*, vol. 58, no. 2, pp. 189–207, 2003.

[20] N. Tatarchuk, J. Shopf, and C. DeCoro, "Real-time isosurface extraction using the gpu programmable geometry pipeline," in *ACM SIGGRAPH 2007 courses*, ACM, 2007, pp. 122–137.

[21] E. S. Lengyel and J. D. Owens, *Voxel-based terrain for real-time virtual simulations*. University of California at Davis, 2010.

[22] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature sensitive surface extraction from volume data," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 2001, pp. 57–66.

[23] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," in *ACM transactions on graphics (TOG)*, ACM, vol. 21, 2002, pp. 339–346.

[24] S. Schaefer, T. Ju, and J. Warren, "Manifold dual contouring," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 610–619, 2007.

[25] T. Lewiner, V. Mello, A. Peixoto, S. Pesco, and H. Lopes, "Fast generation of pointerless octree duals," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1661–1669, 2010.

[26] C. E. V. Muniz, A. Montenegro, M. Lage, and C. N. Vasconcelos, "Polygonal mesh extraction from digital voxel art," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*, IEEE, 2013, pp. 171–178.

[27] G. M. Nielson, G. Graf, R. Holmes, A. Huang, and M. Phielipp, "Shrouds: Optimal separating surfaces for enumerated volumes," in *VisSym*, vol. 3, 2003, pp. 75–84.

[28] F. Labelle and J. R. Shewchuk, "Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles," in *ACM Transactions on Graphics (TOG)*, ACM, vol. 26, 2007, p. 57.

[29] D. Boltcheva, M. Yvinec, and J.-D. Boissonnat, "Feature preserving delaunay mesh generation from 3d multi-material images," in *Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 1455–1464.

[30] J. R. Bronson, J. A. Levine, and R. T. Whitaker, "Lattice cleaving: Conforming tetrahedral meshes of multimaterial domains with bounded quality," in *Proceedings of the 21st International Meshing Roundtable*, Springer, 2013, pp. 191–209.

[31] N. Faraj, J.-M. Thiery, and T. Boubekeur, "Multi-material adaptive volume remesher," *Compurer and Graphics Journal (proc. Shape Modeling International 2016)*, 2016.

[32] L. He and S. Schaefer, "Mesh denoising via l_0 minimization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 64, 2013.

[33] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted l_1-analysis compressed sensing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 2, p. 18, 2014.

[34] X. Wu, J. Zheng, Y. Cai, and C.-W. Fu, "Mesh denoising using extended rof model with l1 fidelity," *Computer Graphics Forum*, vol. 34, no. 7, pp. 35–45, 2015.

[35] H. Zhang, C. Wu, J. Zhang, and J. Deng, "Variational mesh denoising using total variation and piecewise constant function space," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 7, pp. 873–886, 2015.

[36] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," in *Computer Graphics Forum*, Wiley Online Library, vol. 34, 2015, pp. 23–34.

[37] K. Hildebrandt and K. Polthier, "Constraint-based fairing of surface meshes," in *Symposium on Geometry Processing*, Citeseer, 2007, pp. 203–212.

[38] Y. Zhang, C. Bajaj, and G. Xu, "Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow," *Communications in Numerical Methods in Engineering*, vol. 25, no. 1, pp. 1–18, 2009.

[39] R. Klette and A. Rosenfeld, *Digital geometry: Geometric methods for digital picture analysis*. Elsevier, 2004.

[40] D. Coeurjolly, J.-O. Lachaud, and J. Levallois, "Multigrid Convergent Principal Curvature Estimators in Digital Geometry," en, *Computer Vision and Image Understanding*, vol. 129, no. 1, pp. 27–41, Jun. 2014.

[41] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1765–1774, 2012.

[42] D. Coeurjolly, M. Foare, P. Gueth, and J.-O. Lachaud, "Piecewise smooth reconstruction of normal vector field on digital data," *Computer Graphics Forum, Pacific Graphics 2016 Proceedings*, Proc. Pacific Graphics 2016, vol. 35, no. 7, Sep. 2016.

[43] A. Chica, J. Williams, C. Andujar, P. Brunet, I. Navazo, J. Rossignac, and A. Vinacua, "Pressing: Smooth isosurfaces with flats from binary grids," *Computer Graphics Forum*, vol. 27, no. 1, pp. 36–46, 2008, ISSN: 01677055.

[44] C. Andujar, P. Brunet, A. Chica, I. Navazo, J. Rossignac, and A. Vinacua, "Computing maximal tiles and application to impostor-based simplification," in *Computer Graphics Forum*, vol. 23, Amsterdam: North Holland, 2004, pp. 401–410.

[45] L. J. Latecki, "3d well-composed pictures," *Graphical Models and Image Processing*, vol. 59, no. 3, pp. 164–172, 1997, ISSN: 1077-3169.

[46] G. T. Herman, *Geometry of digital spaces*. Springer Science & Business Media, 2012.

[47] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 121–146, 2005.

[48] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun, "Voronoi-based variational reconstruction of unoriented point sets," in *Symposium on Geometry processing*, vol. 7, 2007, pp. 39–48.

[49] H. Pottmann, J. Wallner, Q. Huang, and Y. Yang, "Integral invariants for robust geometry processing," *Computer Aided Geometric Design*, vol. 26, no. 1, pp. 37–60, 2009, ISSN: 01678396.

[50] Q. Mérigot, M. Ovsjanikov, and L. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 6, pp. 743–756, 2011.

[51] H. Pottman, A. Asperl, M. Hofer, A. Kilian, and D. Bentley, *Architectural geometry*. Bentley Institute Press Exton, 2007, vol. 724.

[52] J. Lachaud and B. Thibert, "Properties of gauss digitized shapes and digital surface integration," *Journal of Mathematical Imaging and Vision*, vol. 54, no. 2, pp. 162–180, 2016.
[53] J.-O. Lachaud, D. Coeurjolly, and J. Levallois, "Robust and Convergent Curvature and Normal Estimators with Digital Integral Invariants," in *Modern Approaches to Discrete Curvature*, ser. Lecture Notes in Mathematics, P. R. Laurent Najman, Ed., vol. 2184, Springer-Verlag, 2017.
[54] G. Guennebaud, B. Jacob, *et al.*, *Eigen v3*, http://eigen.tuxfamily.org, 2010.
[55] The DGtal Project, *DGtal*, D. Coeurjolly and J.-O. Lachaud, Eds., 2010. [Online]. Available: https://dgtal.org.
[56] ——, *DGtal*, D. Coeurjolly and J.-O. Lachaud, Eds., version 1.1, Oct. 2020. [Online]. Available: https://github.com/DGtal-team/DGtal/releases/tag/1.1.
[57] P. Alliez, C. Jamin, L. Rineau, S. Tayeb, J. Tournois, and M. Yvinec, "3D mesh generation," in *CGAL User and Reference Manual*, 4.11, CGAL Editorial Board, 2017. [Online]. Available: http://doc.cgal.org/4.11/Manual/packages.html%5C#PkgMesh_3Summary.
[58] J. S. Mitchell, D. M. Mount, and C. H. Papadimitriou, "The discrete geodesic problem," *SIAM Journal on Computing*, vol. 16, no. 4, pp. 647–668, 1987.

APPENDIX

We detail the proof of Proposition 2 (self-intersection free reconstruction).

Proof. We assume, without loss of generality, that the grid step h is 1. We prove it for arbitrary meshes $\hat{P}$, hence it will hold for the optimal mesh P^* since it is a minimum in a compact space. For any vertex $\mathbf{p}_i$ of P, let C_i^ϵ be its associated cube, i.e. $C_i^\epsilon := \{\mathbf{q} \in \mathbb{R}^3, \|\mathbf{q} - \mathbf{p}_i\|_\infty \leq \frac{1-\epsilon}{2}\}$.

Using the increasing property of the convex hull function $\mathrm{Conv}(\cdot)$, for an arbitrary face f_k bordered by vertices $(\hat{\mathbf{p}}_{i_1}, \hat{\mathbf{p}}_{i_2}, \hat{\mathbf{p}}_{i_3}, \hat{\mathbf{p}}_{i_4})$ of $\hat{P}$, we have $f_k \subset \mathrm{Conv}(\{\hat{\mathbf{p}}_{i_1}, \hat{\mathbf{p}}_{i_2}, \hat{\mathbf{p}}_{i_3}, \hat{\mathbf{p}}_{i_4}\}) \subset \mathrm{Conv}(C_{i_1}^\epsilon \cup C_{i_2}^\epsilon \cup C_{i_3}^\epsilon \cup C_{i_4}^\epsilon) =: Q_k$, where the first inclusion comes from rule (i), while the second inclusion comes from rule (ii), which implies $\hat{\mathbf{p}}_i \in C_i^\epsilon$.

The face of P, say f_k', corresponding to the face f_k of $\hat{P}$ is an axis-aligned unit quad that is included in the interior of the four cubes $C_{i_1}^0 \cup C_{i_2}^0 \cup C_{i_3}^0 \cup C_{i_4}^0$. We write $f_k' \subset R_k := \mathrm{Int}(C_{i_1}^0 \cup C_{i_2}^0 \cup C_{i_3}^0 \cup C_{i_4}^0)$. Since for any j, $C_j^\epsilon \subset C_j^0$, we conclude that $f_k \subset Q_k \subset R_k$. By definition of digital surfaces, for two faces f_k' and f_j' of P, we have $R_k \cap R_j \neq \emptyset$ if and only if the faces f_k and f_j share an edge or a vertex. By the above relation ($f_k \subset Q_k \subset R_k$), the two faces f_k and f_j may only intersect if they share an edge or a vertex. Since they obviously intersect when they share a vertex or an edge, it is a necessary and sufficient condition.

Last, the fact that a non-empty intersection between two faces f_k and f_j is exactly their common edge or common vertex comes from the fact that this is the case for the tetrahedra $\mathrm{Conv}(f_k)$ and $\mathrm{Conv}(f_j)$. □

David Coeurjolly David Coeurjolly graduated from the Ecole Normale Superieure and the Université Claude Bernard of Lyon, France, in 2000 and received the Ph.D. Degree in digital geometry in 2002 from the Université Lumière Lyon 2. In 2003, he obtained a permanent research position (Chargé de Recherche CNRS) in the LIRIS laboratory, CNRS UMR 5205 Lyon. In 2011, he got a Senior tenured researcher position (Directeur de Recherche CNRS) at the same institute. He led the m2DisCo research team in LIRIS from October 2011 to January 2015. From January 2017, he is director of the Fédération Informatique de Lyon (CNRS). In 2009, he received a Bronze Medal from CNRS that recognized his activities in Digital Geometry. His present research interests include geometry processing, digital and computational geometry, Monte Carlo rendering and computer graphics.

Jacques-Olivier Lachaud Jacques-Olivier Lachaud graduated from ENSIMAG engineering school in Computer Science in 1994 and received a Ph.D. degree in computer science from Joseph Fourier University (Grenoble, France) in 1998. He is currently a Professor of Computer Science at University Savoie Mont Blanc (Chambéry, France) and works in the laboratory of mathematics (LAMA). His research interests are in image segmentation and analysis, geometry processing, more specifically variational models, digital geometry and topology, and discrete calculus. He has written more than thirty papers in international journals and about sixty papers in international conferences on these topics.

Pierre Gueth Pierre Gueth graduated from the Ecole Normale Supérieure and received his Ph.D. in 2011 from in the Institut National des Sciences Appliquées of Lyon, France. After spending a few years as a post dostoral reseacher and research engineer at LIRIS (2013-2016) where he worked on digital and differential geometry, he has joined Adobe 3D&I in 2019 and works on topics related to geometry since then.