

# A Simple Discrete Calculus for Digital Surfaces<sup>\*</sup>

David Coeurjolly<sup>1</sup>[0000-0003-3164-8697] and Jacques-Olivier  
Lachaud<sup>2</sup>[0000-0003-4236-2133]

<sup>1</sup> Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, Villeurbanne, France  
david.coeurjolly@liris.cnrs.fr

<sup>2</sup> Université Savoie Mont Blanc, CNRS, LAMA, F-73000 Chambéry, France  
jacques-olivier.lachaud@univ-smb.fr

**Abstract.** Computing differential quantities or solving partial derivative equations on discrete surfaces is at the core of many geometry processing and simulation tasks. For digital surfaces in  $\mathbb{Z}^3$  (boundary of voxels), several challenges arise when trying to define a discrete calculus framework on such surfaces mimicking the continuous one: the vertex positions and the geometry of faces do not capture well the geometry of the underlying smooth Euclidean object, even when refined asymptotically. Furthermore, the surface may not be a combinatorial 2-manifold even for discretizations of smooth regular shape. In this paper, we adapt a discrete differential calculus defined on polygonal meshes to the specific case of digital surfaces. We show that this discrete differential calculus accurately mimics the continuous calculus operating on the underlying smooth object, through several experiments: convergence of gradient and weak Laplace operators, spectral analysis and geodesic computations, mean curvature approximation and tolerance to non-manifold locii.

**Keywords:** Discrete calculus, differential operators, digital surface.

## 1 Introduction

In many geometry processing and simulation applications, solving variational problems or simulating partial differential equations on the object boundary can be a critical step. Involved shapes are generally represented as discrete meshes. It is thus necessary to have a calculus that operates consistently and accurately on these discrete objects. For embedded graphs or triangular manifold meshes, finite element methods [1,2] or discrete exterior calculus (DEC) [7,8] have played a crucial role in many applied mathematics and geometry processing applications. From a broad perspective, such models provide consistent differential operators to process scalar, vector or tensor functions on meshes or embedded graphs (consistency given by satisfying Stokes' theorem for all discrete elements). These frameworks induce several convergence results for PDE solutions, but with strong hypotheses on the discrete-continuous mapping [10]. On generic non-triangular

---

<sup>\*</sup> This work has been partly funded by CoMEDiC ANR-15-CE40-0006 research grant.

meshes, Virtual Element Methods [20], or ad hoc operators exist [6]. However, these models assume a Euclidean embedding *interpolating* the smooth manifold.

Digital objects and surfaces corresponds to discrete approximation of continuous objects through a discretization process [11], or to partitions in images. In terms of perturbation and stability, digital surfaces, made of isothetic unit squares in  $\mathbb{Z}^3$ , are very specific: vertices *do not interpolate* the continuous object, and geometric normals poorly approximate the continuous normal bundle. Even worse, the primal quad surface may not be a combinatorial 2-manifold. To design stable geometric estimators, a key ingredient is the digitization grid step  $h$  used to represent objects. Hence, a multigrid process, as a function of  $h$ , can be designed to relate a digital surface to the underlying smooth surface [14]. Stable geometric estimators with convergence properties can then be obtained (i.e. the estimation converges to the expected one on the smooth manifold as  $h$  tends to zero) for various quantities: surface area [14], curvature tensor [12], or even higher order functional such as the Laplace-Beltrami operator [3].

A *combinatorial* DEC can be constructed on digital surfaces, but their specific geometry makes it poorly reflects the continuous calculus. This article presents a *new discrete calculus framework dedicated to digital surfaces*, which relies on two ingredients: (i) a *convergent normal vector field*  $\mathbf{u}$  (e.g. the integral invariant normal estimator [12]), which is used to correct the embedding of (ii) a *polygonal differential calculus model* of de Goes *et al.* [6]. Several methods exploits this idea of correcting the embedding with a normal vector field [14,3,13]. Mercat [16] follows this idea with a theory of conformal parametrization and differential operators for digital surfaces restricted to combinatorial 2-manifolds. Our proposal shares some ideas with these works and defines a calculus on generic digital surfaces with a simple per-face construction of the operators.

The paper is organized as follows. We first describe the operator construction from [6] (§2). §3 describes how to correct the geometrical embedding of the surface elements. Finally, we evaluate the performance of the framework on various variational problems (§4).

## 2 Polygonal differential calculus

We focus here on the formulation proposed by de Goes *et al.* for polygonal surfaces embedded in  $\mathbb{R}^3$  [6]. It defines differential operators per face without assumption on the face geometry, which could be non-convex or even non-planar.

Let  $\mathcal{M}$  be a mesh with vertices  $\mathbf{X}$  and faces  $\mathbf{F}$ . For a given face  $f$  with  $n_f$  vertices, we denote by  $\mathbf{X}_f$ , the vertex positions encoded as a  $n_f \times 3$  matrix (the  $i$ -th row corresponds to the position of the vertex  $i$  of the face  $f$ ). In this calculus, we consider scalar functions  $\phi$  defined on vertices of  $\mathcal{M}$  (see Figure 1–(a–b)). For the face  $f$ , we denote by  $\phi_f$  the restriction of  $\phi$  to the face vertices, represented as a vector of size  $n_f$ . As all discrete differential estimators will be linear in the vertex positions and the scalar function values defined at vertices, the matrix representation will easily combine operators with matrix products. For instance, the centroid of a face is given by  $\mathbf{c}_f := \frac{1}{n_f} \mathbf{X}_f^t \mathbf{1}_f$ , with  $\mathbf{1}_f = (1 \dots 1)^t$  of size  $n_f$ .

First per face local operators are constructed, starting from the gradient. To cope with (non-planar) polygonal faces, the weak form of the gradient is sought (*i.e.*  $\int_f \nabla \phi(\mathbf{x}) d\mathbf{x}$  in the continuous setting) leading to a constant gradient integrated per face. Solving the integral on the discrete structure would require an interpolation scheme on the non-planar face. The key ingredient is to focus on the co-gradient  $\nabla \phi^\perp$  that leads to a simpler expression of its weak form using Stoke's theorem:

$$\int_f \nabla \phi^\perp(\mathbf{x}) d\mathbf{x} = \oint_{\partial f} \phi(\mathbf{x}) \mathbf{t}(\mathbf{x}) d\mathbf{x},$$

$\mathbf{t}(\mathbf{x})$  being the unit tangent vector at  $\mathbf{x} \in \partial f$ . When discretizing this form for a polygonal face  $f$ ,  $\mathbf{t}(\mathbf{x})$  corresponds to the (normalized) edge vectors. Now, let us consider a linear function  $\phi_f = \mathbf{X}_f \mathbf{s} + \mathbf{1}_f r$  with  $\mathbf{s} \in \mathbb{R}^s$  and  $r \in \mathbb{R}$ , integrating  $\phi_f$  on an edge correspond to averaging the function value at the endpoints. Hence, the discrete operator for the (integrated) co-gradient is the  $3 \times n_f$  matrix

$$\mathbf{G}_f^\perp := \mathbf{E}_f^t \mathbf{A}_f,$$

$\mathbf{E}_f$  being the matrix encoding the face edges, and  $\mathbf{A}_f$  the operator that averages two consecutive vertex values. To be explicit,  $\mathbf{A}_f$  is the  $n_f \times n_f$  matrix with  $\mathbf{A}^{i,i} = 1/2$ ,  $\mathbf{A}^{i,i+1} = 1/2$  (zero otherwise), and  $\mathbf{D}_f$  denotes the  $n_f \times n_f$  derivative matrix with  $\mathbf{D}^{i,i} = -1$ ,  $\mathbf{D}^{i,i+1} = 1$  (zero otherwise). So  $\mathbf{E}_f = \mathbf{D}_f \mathbf{X}_f$ .

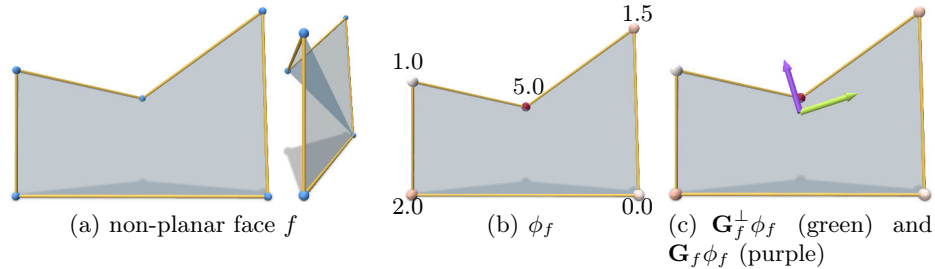
Given a vector  $\phi_f$ ,  $\mathbf{E}_f^t \mathbf{A}_f \phi_f$  is a vector of size 3 corresponding to the Euclidean embedding of the integrated co-gradient of  $\phi_f$  on  $f$ . For a polygonal face  $f$ , the vector area  $\mathbf{a}_f$  can be defined as [17]:  $\mathbf{a}_f := \frac{1}{2} \sum_{v_i \in f} \mathbf{x}_{v_i} \times \mathbf{x}_{v_{i+1}}$ .

The normal vector to  $f$  is simply  $\mathbf{n}_f := \mathbf{a}_f / a_f$ , with  $a_f := \|\mathbf{a}_f\|_2$ . Now, we can finally get the expression of the gradient as  $\nabla \phi^\perp = [\mathbf{n}(\mathbf{x})] \nabla \phi$ , where  $[\mathbf{n}]$  is the  $\pi/2$  rotation matrix such that  $[\mathbf{n}]\mathbf{q} = \mathbf{n} \times \mathbf{q}$  for any 3d vector  $\mathbf{q}$ :

$$\mathbf{G}_f := \frac{1}{a_f} [\mathbf{n}_f] \mathbf{E}_f^t \mathbf{A}_f.$$

The gradient operator is a  $3 \times n_f$  matrix outputting a 3d vector in 3d when applied to a scalar vector  $\phi_f$  (see Figure 1-(c)). Following such per-face construction, de Goes *et al.* define several differential operators listed in Table 1 and represented as matrices acting on scalar functions, vectors, or discrete differential forms (see [7,8] for an introduction). We do not discuss the rationale and the construction of each operator, please refer to [6] for details.

We conclude with the presentation of the Laplace-Beltrami operator  $\Delta$ , a fundamental tool in many geometry processing applications [15]. To solve global PDE involving this operator (*e.g.* for instance solving a Laplace or Poisson problem  $\Delta u = f$ ), it is interesting to aggregate the per-face operators into a global operator  $\mathbf{L}$ . We simply sum up the  $\mathbf{L}_f$  matrices with a global indexing of the vertices. As every  $\mathbf{L}_f$  matrix is negative semi-definite, so is the global Laplace-Beltrami operator. Furthermore, the global operator is very sparse. In opposition with [6], we define it with a negative sign, since usually both the Laplacian and the Laplace-Beltrami are negative operators (with negative eigenvalues).



**Fig. 1.** Illustration of a non-planar face  $f$  (a) equipped with a vertex valued scalar function  $\phi_f$  (b), and the co-gradient and gradient vectors of  $\phi_f$  (c)

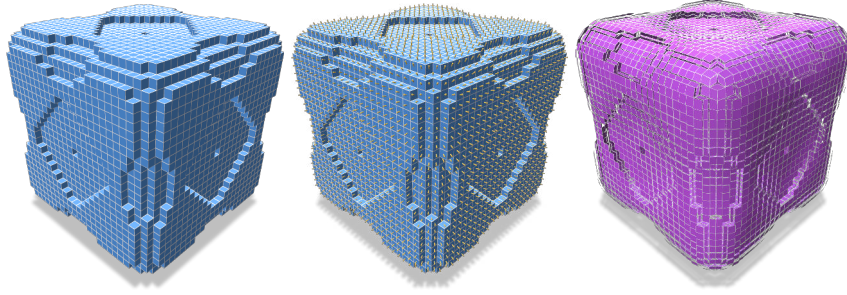
We do not go further into the details of this approach, several results are given in [6] to justify the construction of these operators. For short, they match with classical discrete exterior calculus or finite element ones for triangular meshes [7], and for the polygonal case, they resemble to Virtual Element Methods operators for 2d structures embedded in  $\mathbb{R}^3$  [19]. However, all these relationships between discrete calculus operators and their continuous counterpart on the smooth object only make sense when the discrete structure *interpolates the smooth object*, with a *close normal vector field*. Applying this calculus as is on digital surfaces provide poor results. We need to adapt the surface embedding to correct it.

Operator	Size	Description
$\mathbf{V}_f := \mathbf{E}_f(\mathbf{I}_{3 \times 3} - \mathbf{n}_f \mathbf{n}_f^t)$	$n_f \times 3$	flat operator that maps a vector to a 1-form.
$\mathbf{U}_f := \frac{1}{a_f} [\mathbf{n}_f] (\mathbf{B}_f^t - \mathbf{c}_f \mathbf{1}_f^t)$	$3 \times n_f$	sharp operator that maps a 1-form to a 3d vector.
$\mathbf{P}_f := \mathbf{I}_{n_f \times n_f} - \mathbf{V}_f \mathbf{U}_f$	$n_f \times n_f$	projection operator acting on a 1-form.
$\mathbf{M}_f^0 := \frac{a_f}{n_f} \mathbf{I}_{n_f \times n_f}$	$n_f \times n_f$	inner product for discrete 0-forms (functions).
$\mathbf{M}_f^1 := a_f \mathbf{U}_f^t \mathbf{U}_f + \lambda \mathbf{P}_f^t \mathbf{P}_f$	$n_f \times n_f$	inner product for discrete 1-forms (for some $\lambda > 0$ ).
$\text{Div}_f := -\mathbf{D}_f^t \mathbf{M}_f^1$	$n_f \times n_f$	integrated divergence operator from a 1-form.
$\mathbf{L}_f := -\mathbf{D}_f^t \mathbf{M}_f^1 \mathbf{D}_f$	$n_f \times n_f$	weak Laplace-Beltrami operator.

**Table 1.** Summary of local per-face operators of polygonal calculus [6].

### 3 Projected digital surface embedding

As discussed in the introduction, the classical approach is to relate a digital object to its continuous counterpart through the Gauss digitization process [11]. In this setting, many multigrid convergence results have been obtained for various integral and differential quantities such as the length in 2d [4], the surface area in 3d [14], the curvature tensor [12], or the Laplace-Beltrami operator [3]. Among these techniques, the convergent estimation of the normal vector bundle is the cornerstone of many followup results (*e.g.* [14,3]).



**Fig. 2.** Input digital surface (left), its estimated normal vectors using [12] (middle), projections of the face vertices onto their respective estimated tangent plane (right).

In the polygonal calculus described above, a key ingredient is that operators are first defined *per-face*, and later summed up globally with the vertex indexing. This suggests that correcting the embedding of the face vertices during the local construction of the operators would lead to globally corrected operators. As a first-order approximation, we propose to define our *projected polygonal calculus* by relying on an external estimation of the face tangent space, and by (implicitly) projecting each face vertex onto this tangent plane when constructing the operators. The idea of incorporating external normal vectors to correct the embedding of a discrete structure follows the idea of Lachaud et al [13]. Let  $\{\mathbf{u}_f\}$  be a normal vector field estimated on the digital surface, for instance using the multigrid convergent approach of [12].

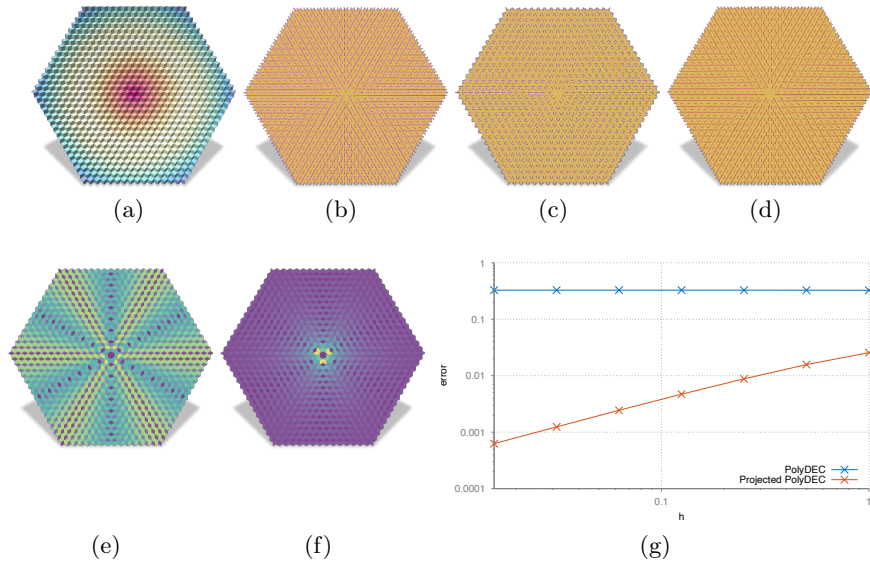
**Definition 1 (Embedding operator).** *The per-face  $f$  projection operator onto the tangent plane orthogonal to  $\mathbf{u}_f$  passing through the origin is the  $3 \times 3$  matrix  $\Pi_f := (\mathbf{I}_{3 \times 3} - \mathbf{u}_f \mathbf{u}_f^t)$ . Recalling that  $\mathbf{X}_f$  is the  $n_f \times 3$  matrix of the vertices position of the face  $f$ , the new positions are given by  $\mathbf{X}_f^* := \mathbf{X}_f \Pi_f$ .*

Note that the intercept does not need to be specified as all differential operators are invariant by translation. For illustration purpose, projected faces have been translated to keep their centroid invariant in Figure 2, but this is meaningless in the calculus. Another observation is that for a given vertex  $\mathbf{v}$  in  $\mathcal{M}$ , its embedding generally differs for its adjacent faces. This follows from the per-face construction of the operators. The global continuity of scalar functions, or their processing (*i.e* their Laplacian) comes from the global indexing of vertices, and global operators as defined above.

## 4 Experiments

We demonstrate the interest of the projected calculus model for the processing of scalar functions on digital surfaces. More precisely, we first evaluate the relevance of the new embedding for the gradient computation of a scalar function. Next, we study the Laplace-Beltrami operator and more advanced processing. All operators are implemented in C++ and available in DGtal [18].

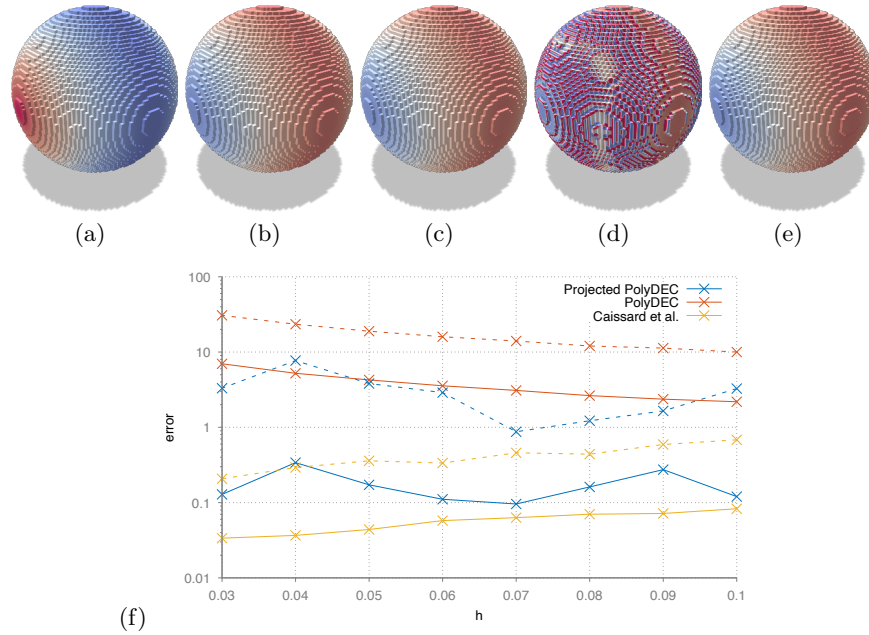
**Gradient of scalar functions.** To evaluate the relevance of including external tangent information in the calculus, let us consider the simple case of the gradient evaluation on a quadratic scalar field defined on a tilted digital plane (with normal vector  $1/3 \cdot (\sqrt{3}, \sqrt{3}, \sqrt{3})^t$ , Fig. 3). In a multigrid setting, we compute the per-face gradient of the scalar function (Euclidean distance in ambient space with respect to the plane center, Fig. 3-(a)), and we compare the expected gradient vector direction (Fig. 3-(b)), to the estimated gradient using the classical polygonal calculus (Fig. 3-(c)), and the projected one (Fig. 3-(d)). Gradient vectors are illustrated from their projection onto the Euclidean plane. The error metric used here is the  $l_2$  norm of the difference between estimated and expected normalized gradient vectors. Figures 3-(e) and (f) detail relative errors. In Figure 3-(g), we have considered a decreasing grid step  $h$  from 1 to  $1.5 \times 10^{-2}$  (3 748 021 faces), demonstrating multigrid convergence of the gradient direction for the projected calculus.



**Fig. 3.** Convergence of the gradient operator: Input scalar function on a tilted plane (a), expected normalized gradient projected onto the Euclidean plane (b), gradient vectors using the original polygonal calculus with many aliasing artifacts (c), gradient vectors using the projected calculus (d). Figures (e) and (f) details relative errors of gradient estimations (values respectively in  $[0, 0.28]$  and  $[0, 0.62]$ ), and (g) illustrates  $l_2$  norm of the error when decreasing the grid step  $h$  (from right to left).

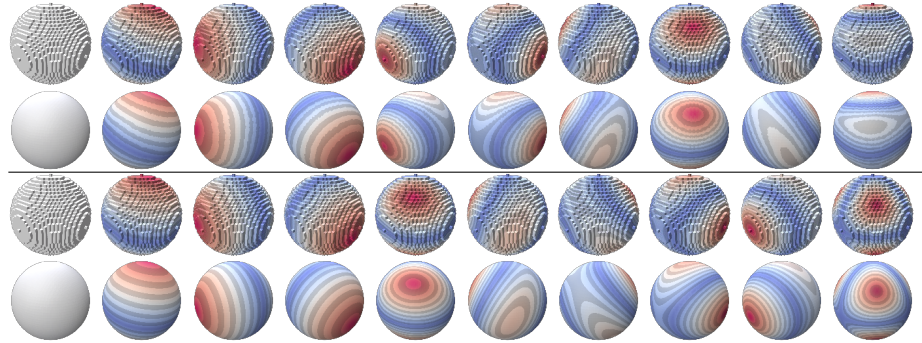
**Laplacian of a function.** First, let us consider an evaluation of the output of the Laplace-Beltrami function on analytical functions on a domain for which we already know the expected Laplacian of that function. Following the setting in [3], we evaluate  $\mathbf{L}$  on a digital sphere with the simple quadratic function  $\phi(\mathbf{x}) := \mathbf{x}_x^2$  on its boundary, and expect  $\Delta\phi(\mathbf{x}) = 2 - 6 \cos^2(\psi)^2(1 - \cos(\theta)^2)$  in

spherical coordinates. Note that Caissard *et al.* [3] achieve strong consistency of the operator (*i.e.* pointwise convergence of the Laplacian values) at the price of a convolution process on the surface elements, leading to a quadratic algorithm for the operator construction, and a non-sparse operator. Clearly it is unlikely that the purely local construction of our projected polygonal calculus provides pointwise convergence. We observe nevertheless that the projection process offers reasonable estimations while being purely local (see Figure 4).



**Fig. 4.** Given a function  $\phi(\mathbf{x})$  in (a) and its analytical Laplacian (b), we illustrate the result of the pointwise convergence heat kernel based Laplace-Beltrami operator [3] in (c). In (d) we have the results of the original local polygonal calculus operator, and in (e) the result of our projected one. In (f) we have performed a multigrid convergence test with decreasing gridstep  $h$ . Dashed lines correspond to the maximum absolute error when estimating  $\Delta f$ , the solid ones correspond to mean absolute errors.

**Spectral analysis.** We evaluate the stability of the spectrum of the pointwise Laplace-Beltrami operator  $(\mathbf{M}^0)^{-1}\mathbf{L}$ . Eigenvalues and eigenvectors have a fundamental role in many geometry processing applications [15] as they define a (spectral) basis to represent scalar functions on surfaces. Figure 5 compares the first eigenvectors (corresponding to the largest eigenvalues) of the original polygonal calculus Laplace-Beltrami operator, to the ones we propose. Our embedding operator has a clear positive impact on the smoothness of basis functions.



**Fig. 5.** First ten eigenvectors of the Laplace-Beltrami operator on a digital sphere (and their projection onto the Euclidean sphere): top two rows correspond to the original polygonal calculus, bottom two rows correspond to the projected polygonal calculus we propose (note that due to proximity of eigenvalues and numerical issues, eigenvectors may not appear in the exact same order).

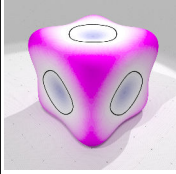
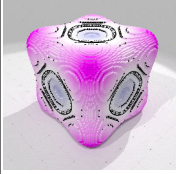
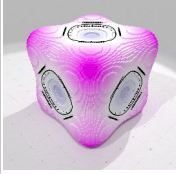
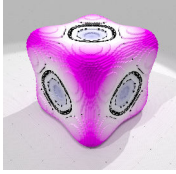
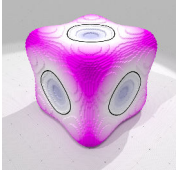
**Mean curvature estimation.** A well-known result related to the Laplace-Beltrami operator is that, if  $\mathbf{x}$  is the function associating to any point of  $\mathcal{M}$  its coordinates, then  $\Delta \mathbf{x} = 2H(\mathbf{x})\mathbf{n}(\mathbf{x})$  with  $H(\mathbf{x})$  the mean curvature at  $\mathbf{x}$  and  $\mathbf{n}(\mathbf{x})$  the normal to  $\mathcal{M}$ . We check this method for computing the mean curvature with our calculus. We compute the pointwise Laplace-Beltrami of coordinates functions  $(X, Y, Z) = \mathbf{x}$  and deduce the mean curvature  $H$  by scalar product with the estimated normal  $\mathbf{u}$ . Otherwise said, the estimated curvature  $\hat{H}$  is

$$\hat{H} = \frac{1}{2}(\mathbf{M}^0)^{-1} [\mathbf{LX} \ \mathbf{LY} \ \mathbf{LZ}] \mathbf{u}^t,$$

where  $\mathbf{u}$  is a  $n_f \times 3$  matrix storing per row the corrected normal to the vertex. To account for the fact that the Laplace-Beltrami is only weakly convergent, we can diffuse for a short time the result to simulate a local integration. Figure 6 illustrates this method for approximating the mean curvature, and also confirms that the calculus must be corrected to get meaningful result.

**Geodesic distance estimation.** To evaluate the proposed calculus on a more complex example, let us consider the geodesic distance estimation problem on digital surfaces. We consider the PDE approach of Crane *et al.* [5] that uses heat diffusion to estimate the geodesic distance function from sources. Sources are defined via a characteristic function  $u_0$  in the domain. The heat method consists in three steps: (i) integrate the heat flow  $\dot{u} = \Delta u$  for some fixed time  $t$  with initial condition  $u(x, 0) = u_0(x)$ , (ii) normalize the gradient vector field  $d := -\nabla u_t / \|\nabla u_t\|_2$ , and (iii) solve the Poisson equation  $\Delta \phi = \nabla \cdot d$  ( $\nabla \cdot$  denotes the divergence operator on vector fields). The scalar function  $\phi - \min_x(\phi)$  is a good approximation of the geodesic distance from the sources to any point [5].



Goursat polynomial surface, digitized at gridstep $h = 0.25$				
Smooth surface	Naive calculus		Projected calculus	
True $H$	Pointwise $\hat{H}$	Diffused $\tilde{H}$	Pointwise $\hat{H}$	Diffused $\tilde{H}$
				
$\ H - \hat{H}\ _2$	0.059	0.051	<b>0.025</b>	<b>0.003</b>
$\ H - \hat{H}\ _\infty$	0.197	0.146	<b>0.155</b>	<b>0.022</b>

**Fig. 6.** Approximation of mean curvature  $H$  with Laplace-Beltrami operator. For the computation of the diffused approximation  $\tilde{H}$ , the diffusion time  $t$  is set to  $0.04 < \frac{1}{6}h$ , so very short. The projected calculus uses integral invariant normal estimator (default parameter  $r = 3h^{1/2}$ ). Projected calculus results are more accurate and stable.

In our discrete calculus framework, the heat diffusion step (i) is obtained with a single backward Euler step for timestep  $t$  solving

$$(u_t - u_0)/t = (\mathbf{M}^0)^{-1}\mathbf{L}u_t \quad \text{or more simply} \quad (\mathbf{M}^0 - t\mathbf{L})u_t = \mathbf{M}^0u_0,$$

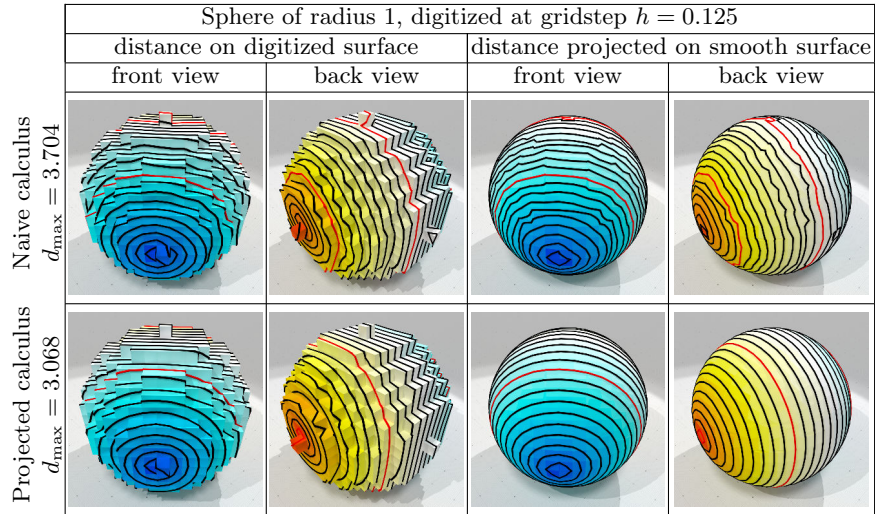
which involves the global mass matrix  $\mathbf{M}^0$  and the weak Laplace-Beltrami  $\mathbf{L}$ .

Then, denoting  $w$  the solution  $u_t$ , step (ii) computes per-face the normalized opposite gradient  $d_f$  of  $w_f$  for face  $f$  as  $d_f := -\mathbf{G}_f w_f / \|\mathbf{G}_f w_f\|_2$ . Bringing back this vector to a one-form with flat operator  $\mathbf{V}_f$ , the per-face weak divergence is then  $\delta_f := \text{Div}_f \mathbf{V}_f d_f$ . Vector  $\delta_f$  is a  $n_f \times 1$  vector associated to face  $f$ , assigning values to each vertex of  $f$ . The global divergence map  $\delta$  simply consists in summing up per vertex the divergence contribution of each face.

Finally, step (iii) computes the geodesic distance function  $\phi$  by solving the Poisson linear system  $\mathbf{L}\phi = \delta$ , which must be shifted by  $-\min_x(\phi)$  so that the result starts from distance 0.

Figure 7 illustrates geodesic distance computations on a digitized sphere, both with naive calculus and projected calculus (we use integral invariant normal estimations with default parameters). As shown by projecting back distances on the true smooth surface, the projected calculus builds more isotropic isodistance lines while being much more numerically accurate.

**Distances on surfaces with boundaries.** If the digital surface has boundaries, then the heat diffusion as it is written follows Neumann boundary conditions: the Laplace-Beltrami operator  $\mathbf{L}$  is built per-face and ignores boundaries. Isodistance lines will tend to be orthogonal to boundaries (see Figure 8 for an example on a half-sphere). As suggested in [5], we compute a second diffusion in step (i) that assumes Dirichlet zero boundary conditions on the digital surface boundaries. This is done by restricting the linear system  $(\mathbf{M}^0 - t\mathbf{L})u_t = \mathbf{M}^0u_0$  to non-boundary vertices, simply by assuming value 0 for  $u$  on boundary vertices.



**Fig. 7.** Distance computation to a source point with geodesic in heat method on a digital sphere. Scale is deep blue (0) to red (max distance) with black isolines every 0.1 and red isolines every 1. The expected max distance is  $\pi$ . Top row displays distances computed with naive polygonal calculus. Bottom row displays distances computed with projected calculus with normals  $\mathbf{u}$  estimated with integral invariant method. All computations made with parameter  $\lambda = 0.25$  and heat diffusion time  $t = h^2$ .

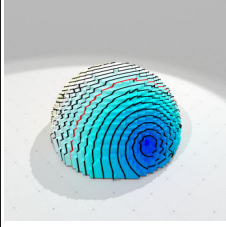
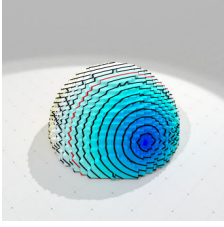
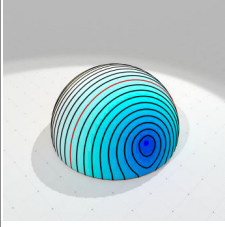
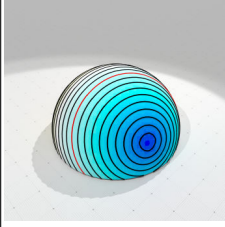
We then average the two solutions to define the vector  $w$  that is given to step (ii) of the heat method. Figure 8 confirms the soundness of this approach.

**Processing on non-manifold surfaces.** Since every calculus operator is defined per-face, they are quite oblivious to non-manifold edges and vertices of the digital surface. We illustrate this fact on Figure 9 by computing the geodesic distance to a source point on a digital surface approximating the triple junction of three planes. Not a single line of code is changed.

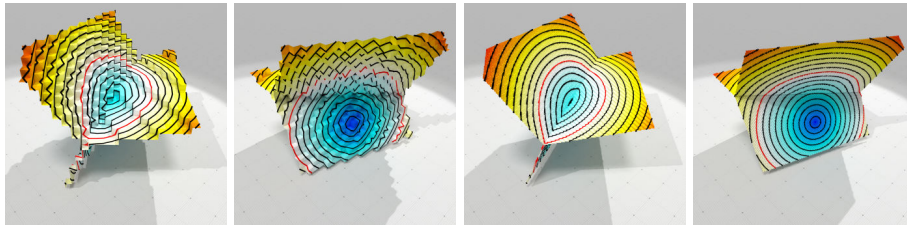
## 5 Conclusion

Our contribution is a simple and easily implementable discrete differential calculus for the processing of scalar functions on digital surfaces. It relies on changing the natural embedding of the digital surface when constructing per face differential operators, with the use of an external, multigrid convergent, normal vector estimation. Many experiments back up the effectiveness of this new calculus.

Although we demonstrate its interest to solve integrated problems such as Poisson problem (and the geodesic distance estimation is a perfect example of this class), the proposed local construction does not achieve pointwise convergence for second order operators like the Laplace-Beltrami one. Interesting challenges include the design of nonlocal operators, similarly to [3] or [9], or the use of higher order embeddings for the digital surface.

Half-sphere of radius 1, digitized at gridstep $h = 0.05$			
distance on digitized surface		distance projected on smooth surface	
			
Solely Neumann b. c.	Mixed Neumann and Dirichlet b. c.	Solely Neumann b. c.	Mixed Neumann and Dirichlet b. c.

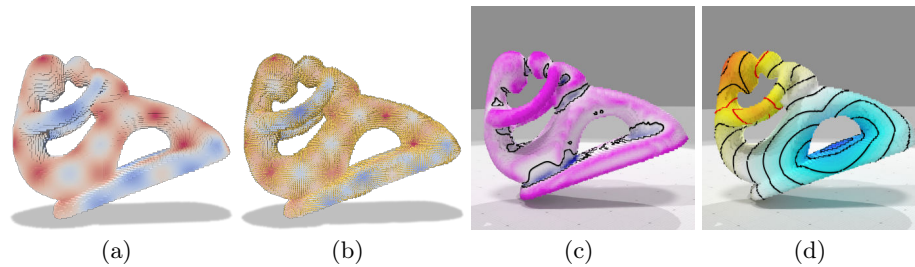
**Fig. 8.** Distance computation to a source point with geodesic in heat method on a digital surface with boundaries (here a half-sphere). Results are more accurate when mixing two heat diffusion solutions, one with Neumann boundary conditions, one with Dirichlet boundary conditions.



**Fig. 9.** Geodesic distance computation on non-manifold surfaces: the projected calculus is quite oblivious to non-manifold parts of the surface, as illustrated on this triple junction surface, digitized coarsely at  $h = 1$  (left) and finely at  $h = 0.125$  (right).

## References

1. Arnold, D.N., Falk, R.S., Winther, R.: Differential complexes and stability of finite element methods i. the de rham complex. In: Compatible spatial discretizations, pp. 23–46. Springer (2006)
2. Arnold, D.N., Falk, R.S., Winther, R.: Finite element exterior calculus, homological techniques, and applications. *Acta numerica* **15**, 1–155 (2006)
3. Caissard, T., Coeurjolly, D., Lachaud, J.O., Roussillon, T.: Laplace–beltrami operator on digital surfaces. *Journal of Mathematical Imaging and Vision* **61**(3), 359–379 (2019). <https://doi.org/10.1007/s10851-018-0839-4>
4. Coeurjolly, D., Klette, R.: A comparative evaluation of length estimators of digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(2), 252–258 (Feb 2004)
5. Crane, K., Weischedel, C., Wardetzky, M.: The heat method for distance computation. *Commun. ACM* **60**(11), 90–99 (Oct 2017). <https://doi.org/gcj3hk>
6. De Goes, F., Butts, A., Desbrun, M.: Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics (TOG)* **39**(4), 110–1 (2020)
7. Desbrun, M., Hirani, A.N., Leok, M., Marsden, J.E.: Discrete exterior calculus. arXiv preprint math/0508341 (2005)



**Fig. 10.** Various computations on a more complex voxel object: (from left to right) the gradients of a scalar function, mean curvature and geodesic distances.

8. Grady, L.J., Polimeni, J.: *Discrete calculus: Applied analysis on graphs for computational science*. Springer (2010)
9. Hildebrandt, K., Polthier, K.: On approximation of the laplace–beltrami operator and the willmore energy of surfaces. In: *Computer Graphics Forum*. vol. 30, pp. 1513–1520. Wiley Online Library (2011)
10. Hildebrandt, K., Polthier, K., Wardetzky, M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata* **123**(1), 89–112 (2006)
11. Klette, R., Rosenfeld, A.: *Digital geometry: Geometric methods for digital picture analysis*. Morgan Kaufmann (2004)
12. Lachaud, J.O., Coeurjolly, D., Levallois, J.: Robust and Convergent Curvature and Normal Estimators with Digital Integral Invariants. In: Laurent Najman, P.R. (ed.) *Modern Approaches to Discrete Curvature*, Lecture Notes in Mathematics, vol. 2184. Springer-Verlag (2017)
13. Lachaud, J.O., Romon, P., Thibert, B., Coeurjolly, D.: Interpolated corrected curvature measures for polygonal surfaces. *Computer Graphics Forum (Proceedings of Symposium on Geometry Processing)* **39**(5) (2020). <https://doi.org/gmt2mq>
14. Lachaud, J.O., Thibert, B.: Properties of gauss digitized shapes and digital surface integration. *Journal of Mathematical Imaging and Vision* **54**(2), 162–180 (2016)
15. Lévy, B., Zhang, H.: Spectral mesh processing. In: *ACM SIGGRAPH 2010 Courses*, pp. 1–312 (2010)
16. Mercat, C.: Discrete complex structure on surfel surfaces. In: *International Conference on Discrete Geometry for Computer Imagery*. pp. 153–164. Springer (2008)
17. Sullivan, J.M.: Curvatures of smooth and discrete surfaces. In: *Discrete differential geometry*, pp. 175–188. Springer (2008)
18. The DGtal Project: DGtal (2010), <https://dgtal.org>
19. Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* **23**(01), 199–214 (2013)
20. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: The hitchhiker’s guide to the virtual element method. *Mathematical models and methods in applied sciences* **24**(08), 1541–1573 (2014)