Computing Homological Information Based on Directed Graphs within Discrete Objects

Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, Pedro Real

HAL Id: hal-01381797

https://hal.archives-ouvertes.fr/hal-01381797

Submitted on 21 Feb 2017

Computing Homological Information Based on Directed Graphs within Discrete Objects

Aldo Gonzalez-Lorenzo*[†], Alexandra Bac*, Jean-Luc Mari* and Pedro Real[†]

*Aix-Marseille Université, CNRS, LSIS UMR 7296 (France)

[†]University of Seville, Institute of Mathematics IMUS (Spain)

Abstract—n-**dimensional discrete objects can be interpreted as cubical complexes which are suitable for the study of their homology groups in order to understand the original discrete object. The classic approach consists in computing the Normal Smith Form of some matrices associated to the cubical complex. Further approaches deal mainly with a pre-processing of the matrices in order to reduce their size.**

In this paper we propose a new approach, initially based on discrete Morse theory, which computes some homological information (Betti numbers and representative cycles) without calculating the Normal Smith Form. It works on any dimension, and it can also be applied to any kind of regular cell complex.

I. INTRODUCTION

Homology computation is a very useful tool for classification and understanding of discrete objects (mainly issued from binary digital images) notably when they have more than three dimensions. It provides a class of descriptors summarizing the basic structure of the considered shape.

Homology deals with the "holes" of an object. Holes can be classified by dimensions: 0-holes correspond to connected components, 1-holes to tunnels and 2-holes to cavities. When computing homology, we usually expect to find the number of these holes (called the Betti numbers) and a representative of each hole (called representative cycle of a homology generator). This is easy to understand up to dimension 3. However, starting from $\mathbb{R}^4$, *torsion* coefficients may arise, which represent holes that are filled if we walk them a determinate number of times. This is difficult to imagine, but it remains a good tool for classifying objects. Closely related to this problem is the choice of the ring of coefficients in the homology. Up to dimension 3, we can choose any. However, in order to be able and handle any dimension, we have chosen the ring of integers $\mathbb{Z}$.

Among other approaches, let us mention the following two, which are closely related to our work: Effective Homology Theory [1] and Discrete Morse Theory [2]. Both of them will be explained in sections III-D and III-E. The former has the advantage that it "controls" the homology because it contains all the information; the latter is very concise and easy to implement. Effective Homology Theory deals with linear maps that should be typically encoded as enormous matrices; Discrete Morse Theory handles only graphs but does not always compute the homology with accuracy.

We aim to find an intermediate solution, avoiding the respective drawbacks of both of these methods and keeping their advantages. Roughly speaking, Discrete Morse Theory gives an approximation of the homology of an object by establishing arrows on it, what can be seen as a spanning forest [3]. In this paper we allow cycles in this "collection of arrows", which is normally forbidden, so that we can compute the exact homological information. Starting from these arrows (an oriented graph), we introduce an algorithm computing the linear maps of Effective Homology. We can say that, instead of using quadratic space for saving these maps, we use linear time and space. Hence there is no magic here. Our contribution mainly lies on how to deal with these cycles.

Our approach consists of 3 main steps. First we transform a discrete object into a cubical complex. Then we establish a sub-graph on it. In 3D, this sub-graph directly gives the Betti numbers. In the third step we extract some linear maps giving more detailed homological information (representative cycles and their relations between them). We limit ourselves to obtain the representative cycles.

This method works in any dimension and on any kind of regular cell complex. In the limited context of this paper, we will only give details in the 3D space and we only consider a concrete kind of cell complexes: the cubical complexes. They are very useful for computing homology on discrete objects. Nevertheless, our approach can be applied to higher dimensions and other kinds of cell complexes without notable changes.

In section II, we review the state of the art related to our work. In section III, we introduce all the necessary definitions for understanding our method. If the reader is not familiarized with these concepts, some references are given. In section IV, we describe our algorithm starting from a minimal example that presents a typical obstacle. Section V shows two typical counter-examples for Discrete Morse Theory. We finish this paper by presenting our conclusion and our future work.

II. PREVIOUS WORKS

We describe in more detail in section III the notions of homology, cubical complexes, their construction from discrete objects and homology computation approaches. Therefore, we mention only some references in the present section.

Two introductory books for homology are [4] and [5].

Homology can be computed over different kinds of structures (cell complexes). In [6] and [7], a kind of cell complex (polyhedral complexes) is built from binary 3D images. In [8] another kind of cell complexes (called cubical complex) is built. The present work is also based on cubical complexes (see section III-B).

There are several approaches to compute the homology of a cell complex. Some of them optimize the calculation of the Normal Smith Form [9], [10]. Others try to reduce the complex [11], [12]. There is a family of methods called Persistent Homology that uses filtrations on the cell complex [13], [14]. A very interesting approach can be found in [15], where the complex is decomposed in order to simplify the computation.

A very clear introduction to Discrete Morse Theory is described in [2]. In [16], [17], [3] some methods for establishing a discrete gradient vector field are given. In [18], [19] an iterative version is given such that the homology is accurately captured.

III. PRELIMINARIES

A. Discrete Objects

A *discrete object* of dimension n is a set of n-dimensional cubes centred on integer coordinates. We will describe it by the set of the coordinates of its elements.
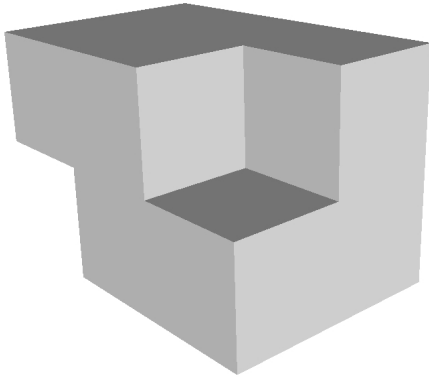

Fig. 1. A 3-dimensional discrete object

The elements of a discrete object are named according to its dimension: pixels (2D), voxels (3D) and doxels (4D).

B. Cubical Complex

This section is derived from [8]. For a deeper understanding of these concepts, the reader can refer to it. An *elementary interval* is an interval of the form $[k, k + 1]$ (nondegenerate) or a set $\{k\}$ (degenerate), also denoted as $[k, k]$, where $k \in \mathbb{Z}$. An *elementary cube* in $\mathbb{R}^n$ is the Cartesian product of n elementary intervals, and the number of nondegenerate intervals in this product is its *dimension*. An elementary cube of dimension q will be called q-cube for short, or even q-cell, since cubical complexes are a special kind of cell complexes.
Given two elementary cubes P and Q, we say that P is a *face* of Q if $P \subset Q$. It is a *primary face* if the difference of their dimensions is 1.
A *cubical complex* is a set of elementary cubes with all of their faces. The *boundary* of an elementary cube is the collection of its primary faces.

Given a discrete object, we can define an associated cubical complex. We will explain this only for dimension 3.
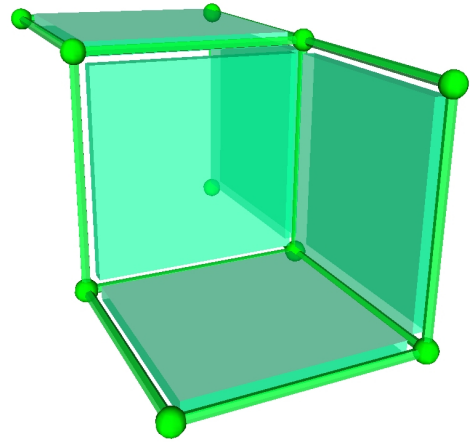

Fig. 2. The cubical complex associated to the discrete object in Fig. 1

- For every voxel (x, y, z), there is a 0-cube $[x, x] \times [y, y] \times [z, z]$;

- For every two 6-adjacent voxels, there is a 1-cube between their associated 0-cubes. For example, if (x, y, z) and $(x+1, y, z)$ belong to the discrete object, the 1-cube is $[x, x + 1] \times [y, y] \times [z, z]$;

- For every four voxels mutually 6-adjacent (each voxels is 6-adjacent to two others), there is a 2-cube;

- Finally, for every eight voxels mutually 6-adjacent (each voxels is 6-adjacent to three others), there is a 3-cube.

Fig. 3 illustrates this construction.

Fig. 3. A discrete object and its associated cubical complex.
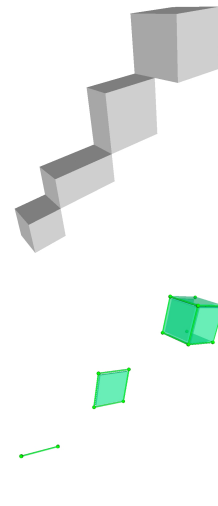
A cubical complex can be completely described by its *Hasse diagram* (also called *cell graph*). It is a directed graph whose vertices are all the elementary cubes, and whose arrows go from each cube to its primary faces. We will strongly use this concept in our approach. In this paper we will usually not make the distinction between the vertices and the elementary

cubes they represent, so we will mix the terms vertex, cube and cell.

C. Chain Complexes and Homology

A *chain complex* (C_*, d_*) is a sequence of groups $C_0, C_1, \ldots$ (called *chain groups*) and homomorphisms $d_1 : C_1 \to C_0, d_2 : C_2 \to C_1, \ldots$ (called *differential* or *boundary operators*) such that $d_{q-1}d_q = 0, \forall q > 0$.
Given a cubical complex, we define its chain complex as follows:

- C_q is the free group generated by the q-cubes of the complex;

- d_q gives the "algebraic" boundary, which is the sum of the primary faces of the q-cubes with a specific coefficient. An explicit formula depending on the elementary intervals is given in [8].

This chain complex should be seen as a sequence of matrices that express the relation of adjacency between the elementary cubes.

The elements of the chain group C_q, which are formal sums of q-cubes, are called q-*chains*. A q-chain x is a cycle if $d_q(x) = 0$, and a boundary if $x = d_{q+1}(y)$ for some $(q+1)$-chain y. We will not always write the subscripts when it is clear from the context. By the property $d_{q-1}d_q = 0$, every boundary is a cycle, but the reverse is not true: a cycle which is not a boundary contains a "hole". The q-th homology group of the chain complex (C_*, d_*) contains the q-dimensional "holes": $H(C)_q = \ker(d_q)/\mathrm{im}(d_{q+1})$. This set is a finitely generated group, so there is a "base" typically formed by the holes of the cubical complex. This group is isomorphic to:

$$\mathbb{Z}^{\beta_q} \times \mathbb{Z}/\lambda_1\mathbb{Z} \times \mathbb{Z}/\lambda_2\mathbb{Z} \times \ldots \times \mathbb{Z}/\lambda_t\mathbb{Z}$$

where each λ_i divides λ_{i+1}. β_q is the q-th *Betti number* and $\lambda_1, \ldots, \lambda_t$ are the *torsion coefficients*. Let us recall that if the cubical complex is in $\mathbb{R}^3$, there are no torsion coefficients.

Given an algebraic boundary operator on a cubical complex, we can add a sign on the arrows of its Hasse diagram. The *sign* of an arrow (u, v) is $\rho(u, v) = \langle d(u), v \rangle$, this is the coefficient of the cell v in the chain $d(u)$. In our context, Hasse diagrams will always have signs on their arrows.

D. Reduction

The classical way of computing the homology of a chain complex is given by a constructive proof of Munkres [4]. It is based on a diagonalization of the matrices of the differential operators. This is practically impossible to perform with large complexes. A solution to reduce the amount of information to compute is the notion of reduction. It is a strong relation between two chain complexes that gives an isomorphism between their homology groups. This is the main tool of the Effective Homology Theory. We typically reduce the initial chain complex to another much smaller.

Formally, a reduction between two chain complexes (C_*, d_*) and (C'_*, d'_*) is a triple of homomorphisms (h_*, f_*, g_*) such that:

- $h_q : C_q \to C_{q+1}$ for every $q \geq 0$

- $f_q : C_q \to C'_q$ is a chain map ($fd = d'f$)

- $g_q : C'_q \to C_q$ is also a chain map ($gd' = dg$)

- $gf = 1 - dh - hd$

- $fg = 1_{C'}$

- $hh = hf = hg = 0$

In our work, we try to find a reduction between the chain complex associated to a discrete object and the smallest possible complex such that the calculation of the homological information becomes trivial. Moreover, we will try to get a minimal description of this reduction, so that we do not need to save it as a sequence of matrices, but as a mere sub-graph of the Hasse diagram.

E. Discrete Morse Theory

Discrete Morse Theory was introduced by Robin Forman as a discretization of the Morse Theory [2]. The main idea is to obtain some homological information by means of a function defined on the complex. This function is equivalent to a discrete gradient vector field and we will rather use this notion.

A *discrete vector field* (DVF) on a cubical complex is a matching on its Hasse diagram, that is a collection of edges such that no two of them have a vertex in common. From a Hasse diagram and a discrete vector field we can define a *Morse graph*: it is a graph similar to the Hasse diagram except for the arrows contained in the matching, which are reverted. These arrows will be called *integral arrows*, and the others, *differential arrows*.

A *V-path* is a path on the Morse graph that alternates between integral and differential arrows. A *discrete gradient vector field* (DGVF for short) is a discrete vector field that does not contain any closed V-path. A *critical vertex* (or critical cell) is a vertex that is not paired by the matching.

One of the main results of Discrete Morse Theory is that the number of critical q-cubes is larger or equal than the q-th Betti number. When they are equal, we say that the DGVF is *perfect*. An *optimal* DGVF contains the least possible number of critical cells. Obviously, every perfect DGVF is optimal. Thus, a DGVF gives an estimation of the Betti numbers without using any algebraic method. We can say that it is a "combinatorial" tool.

Let us point out that starting from a DGVF G, a reduction can be defined. Firstly, let us define a linear operator V which maps vertices containing outward integral arrows to the head of this arrow with its sign. Formally,

$$V(\sigma) = \begin{cases} \langle d(\tau), \sigma \rangle \cdot \tau, & (\sigma, \tau) \in G \\ 0, & \text{if not} \end{cases}$$

Then,

$$h(\sigma) = \sum_{k \geq 0} V(1 - dV)^k(\sigma)$$
$$f(\sigma) = (1 - dh - hd)(\sigma)$$
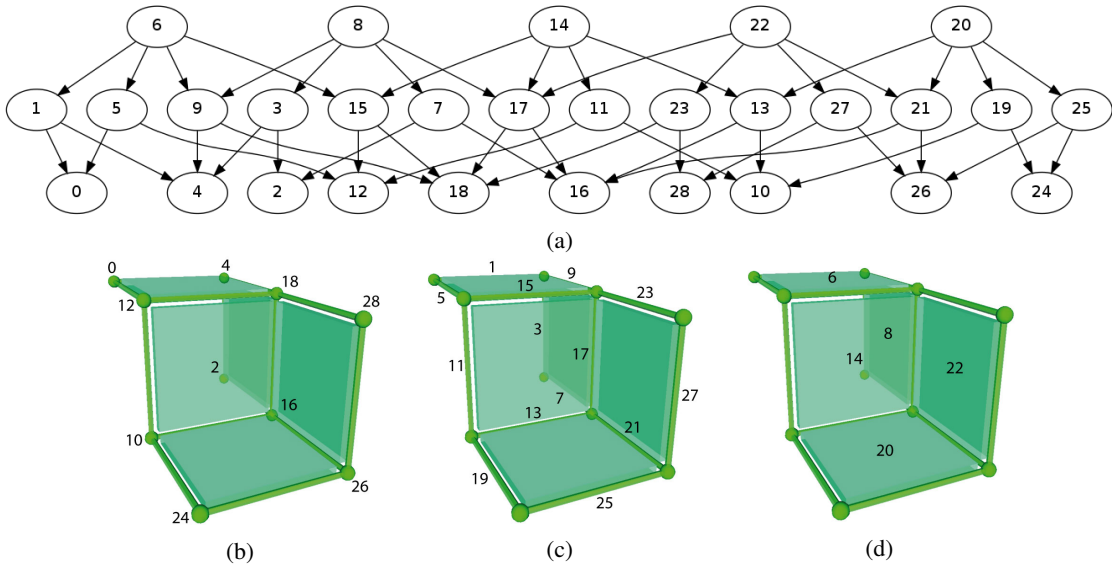$$g(\sigma) = \sigma$$

Fig. 4. (a) Hasse diagram of the cubical complex in Fig. 2. The elementary cubes have been indexed arbitrarely. The correspondence between cells and indexes is shown in figures (b) dimension 0, (c) dimension 1, (d) dimension 2.
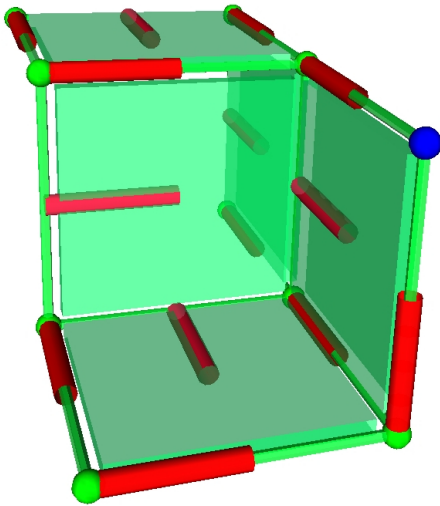

Fig. 5. A DGVF over the complex in Fig. 2. The critical cells are represented in blue. The integral arrows are shown in red and the differential ones are omitted.

Fig. 6. The Morse graph of the DGVF in Fig. 5. Some differential arrows have been omitted for clarity.

Finally, let us point out two nice properties that allow us to compute these maps recursively:

$$h(\sigma) = V(\sigma) + h(1 - dV)(\sigma)$$
$$f(\sigma) = f(1 - dV)(\sigma)$$
(1)

IV. OUR APPROACH

A. Motivation

Discrete Morse Theory approach has a strong interest as it addresses the computation of homology as a purely combinatorial problem rather than an algebraic one. So it does not entail working with possibly huge (but sparse) matrices. Betti numbers, the most relevant homological information, are obtained in a constant 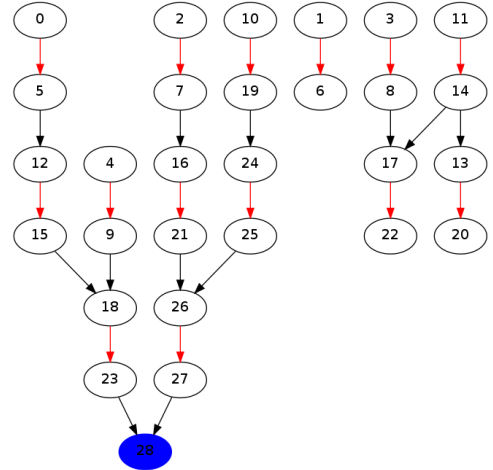time (if there is no torsion); the representative cycles, in linear time. But, most interesting in this approach, is the resulting graph, that can be used to obtain other geometrical information of the complex.

Unfortunately, computing an optimal DGVF (with a minimum number of critical cells) is a NP-hard problem and there are simple complexes for which no DGVF give the exact Betti numbers (as, for instance, the Bing's house or the dunce hat). Our work is an effort to overcome this problem and provides an intermediate solution: our method is still based on Discrete Morse Theory, but we allow directed cycles in the DGVF in order to obtain exact homological information. We will call this kind of DVF a *homological discrete vector field* (HDVF for short). We show that computing the reduction is possible from such DVF and may require solving reduced systems of linear equations in some cases.

B. A First Example

Our contribution consists in an algorithm iteratively *correcting* a non-perfect DGVF. Forman proved that two critical cells can be cancelled if there exists only one V-path between them by reversing it. The present work allows this cancellation even when there are several V-paths, thus creating cycles in the Morse graph. Our algorithm however overcomes this difficulty and computes the associated reduction by carefully considering these cycles.

Let us consider the small scale example of Fig. 7. There are 3 V-paths between the critical 2-cell and 1-cell (drawn in orange).
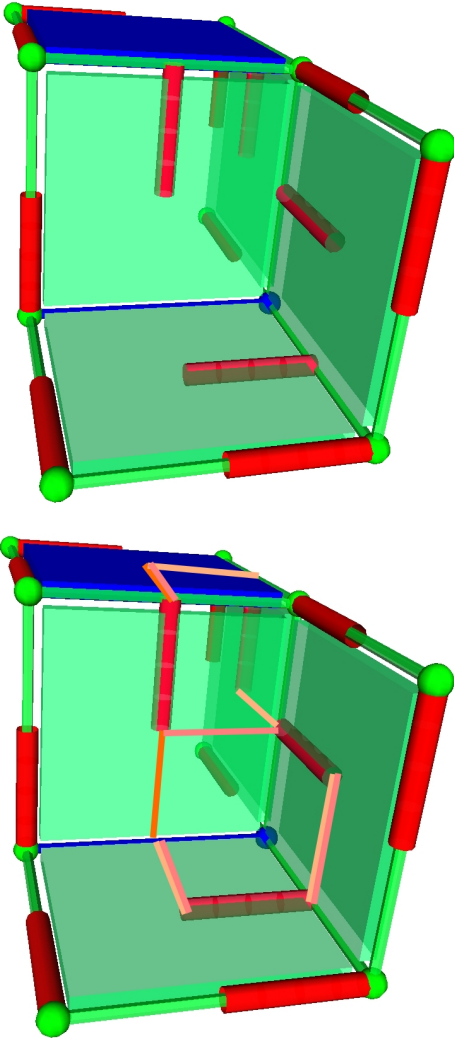

Fig. 7. A DGVF in red over the complex in 2. There are 3 critical cells drawn in blue, one of each dimension

When reversing one of the V-paths (actually the shortest one), we thus obtain the correct number of critical cells. However, we also create two directed cycles. As a consequence, if we try to extract the associated reduction using (1), we obviously enter an infinite loop because of the cycles. As a first step of our approach, we define and compute the *confluence vertices* of the DVF, that is the vertices where directed cycles

merge. Let c be the confluence vertex (drawn in purple). Starting from formula (1) (up) we showed that h can be computed by means of a graph inspection (visited vertices are added to the chain $h(c)$). When vertex c is visited a second time, the process stops following that path. The algorithm eventually produces the recursive equation (2)-left. We then solve this reduced linear system to obtain the value of $h(c)$ (equation (2)-right).

$$h(c) = 2 \cdot h(c) + x \Rightarrow h(c) = -x \qquad (2)$$

where x is a 2-chain. So instead of looping in the recursive formula (1), we stop it by substituting identity (2).
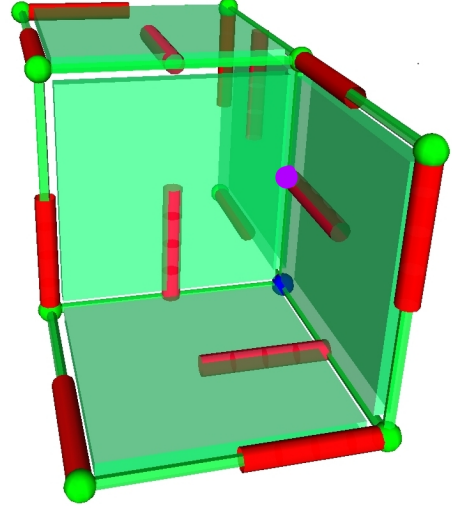

Fig. 8. The DVF after reversing one V-path. The confluence cell is marked by a purple dot.

According to formula (1), the computation of the map f is similar to that of h. As we will see, both algorithms differ in just one case.

C. Detailed Algorithm

Our global process consists in the following steps (see figure 9):

1) Starting from a **discrete object**, we build a **cubical complex** (see section III-B);
2) Computation of the reduction and corrected DVF:
    a) We compute an **initial DGVF** G (various methods exist, see for instance [16], [3] or [17])
    b) Starting from G, we compute a perfect DVF by iterative arrow-reversings. This is the main issue of our algorithm
3) We can then compute the final reduction and hence obtain the representative cycles.

Before going into more details for step 2.b (which is the heart of our work), let us elaborate on the computation of $(1 - dV)$, which is a frequent routine. Algorithm 1 describes its computation from a Morse graph G (let us recall that given an edge (u, v) of the Morse graph, we denote by $\rho(u, v)$ its sign as described in section III-C).
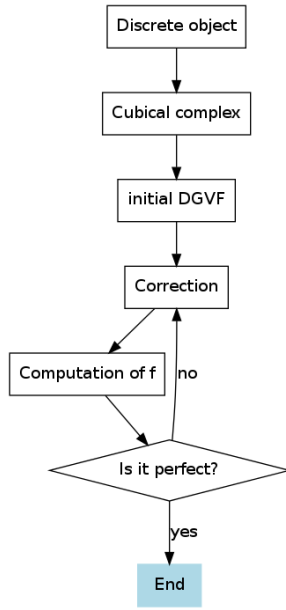
Fig. 9. Overview of our approach.

ALGORITHM 1 Computation of $(1 - dV)(\sigma)$

Require: A Morse graph G and a vertex σ
Ensure: The chain $x = (1 - dV)(\sigma)$
  $x \leftarrow 0$
  if $\exists (\sigma, \Sigma) \in G$ an integral arrow **then**
    for all differential arrow $(\Sigma, \tau) \in G$ **do**
      $x \leftarrow x - \rho(\sigma, \Sigma) \cdot \rho(\Sigma, \tau) \cdot \tau$
    end for
    return x
  else
    return σ
  end if

---

1) Computation of f from a corrected DVF: In order to compute a perfect DVF, our approach is based on iterative arrow-reversing (or DVF correction). This requires calculating the chain fd over the critical cells. This chain must be expressed as the image by f of a chain containing only critical cells. In the sequel, we will say that f is computed *in terms of critical cells*.

The main issue of our algorithm is that after arrow-reversing, G may contain cycles. Hence, computing f over any cell cannot be done at once.

However, we showed that a way to overcome the problem of cycles can be to compute f first over every confluence vertex (see algorithm 2). The algorithm follows the graph G and eventually stops providing a system of linear equations.

ALGORITHM 2 Algorithm for computing f on a confluence vertex

Require: A Morse graph G and a confluence vertex c
Ensure: The chain $x = f(c)$ computed in terms of critical cells
  Q : FIFO queue of cells with coefficients $k \cdot \tau$ ($k \in \mathbb{Z}$, τ : cell)
  $x \leftarrow 0$
  $y \leftarrow (1 - dV)(c)$
  push every cell of y with its coefficient into Q
  while Q is not empty **do**
    pop the first element of Q: $k \cdot \tau$
    if τ is a confluence vertex OR a critical cell **then**
      $x \leftarrow x + k \cdot \tau$
    else
      $y \leftarrow k \cdot (1 - dV)(\tau)$
      push every cell of y with its coefficient into Q
    end if
  end while

Once we have computed f over all the confluence vertices we obtain the following system of linear equations

$$X = AX + B \Rightarrow (I - A)X = B,$$

where the matrix A contains the coefficients of the confluence vertices and B, the coefficients of the critical cells. This system provides a recursive definition of f over the confluence vertices. We solve it to obtain the actual value of f at any confluence vertex.

Starting from these values, algorithm 3 computes the value of f at any vertex σ. Note that it is very similar to algorithm 2.

ALGORITHM 3 Algorithm for computing f on any vertex

Require: A Morse graph G and a vertex σ
Ensure: The chain $x = f(\sigma)$ computed in terms of critical cells
  Q : FIFO queue of cells with coefficients $k \cdot \tau$ ($k \in \mathbb{Z}$, τ : cell)
  $x \leftarrow 0$
  push σ into Q
  while Q is not empty **do**
    pop the first element of Q: $k \cdot \tau$
    if τ is a confluence vertex **then**
      $x \leftarrow x + k \cdot f(\tau)$ with $f(\tau)$ computed in algorithm 2
    else
      if τ is critical **then**
        $x \leftarrow x + k \cdot \tau$
      else
        $y \leftarrow k \cdot (1 - dV)(\tau)$
        push every cell of y with its coefficient in Q
      end if
    end if
  end while

Through the careful handling of confluence vertices, this algorithm overcomes the delicate issue of cycles and computes f in terms of critical cells from a DVF obtained by the iterative correction algorithm described in the next section.

2) Iterative correction of the DVF: As explained previously, the number of critical cells in the Morse graph does not necessarily give Betti numbers, that is, not all Morse graphs

are perfect. Removing unwanted critical cells is the purpose of DVF correction (which actually cancels pairs of critical cells by generalized arrow reversing). Our method consists in running algorithm 4 on every critical cell.

ALGORITHM 4 Cancellation of a critical cell

Require: A Morse graph G and a critical cell σ
Ensure: A corrected Morse graph G'
  $x \leftarrow fd(\sigma)$ in terms of critical cells
  if $\exists \tau$ such that $\langle x, \tau \rangle = \pm 1$ **then**
    compute G' by taking a V-path between σ and τ and reverse it
    compute the new f
  end if

After correcting all critical cells, the DVF is perfect. The number of critical cells equals the Betti numbers.

3) Computation of the final reduction: Beyond the Betti numbers, we can obtain the final reduction (h, f, g). Starting from the perfect DVF obtained previously, we can now compute h. The calculation is analogous to that of f (described in algorithms 2 and 3). The only difference (in both algorithms) is that, when τ is not a confluence vertex, the whole **else** condition simply becomes :

  if $V(\tau) \neq 0$ **then**
    $x \leftarrow x + k \cdot V(\tau)$
    $y \leftarrow k \cdot (1 - dV)(\tau)$
  end if

The explicit value of f is obtained by using the identity $f = 1 - dh - hd$. Last, g is the inclusion map. The representative cycles are obtained by applying $1 - hd$ over the critical cells.

If we do not restrict ourselves to the 3D context, then the homology groups can have torsion coefficients. This case needs more detailed explanations that we will develop in a forthcoming paper.

V. VALIDATION

Our algorithm has been implemented in C++ using the library DGtal [20] together with libraries Eigen [21] and Graphviz [22].

Let us now show the HDVF computed on two spaces that do not admit a perfect DGVF: the Bing's house and the dunce hat [23].

The cubical complex version of the Bing's house has been created by the authors. It contains 60 0-cells, 129 1-cells and 70 2-cells. The first DGVF defined on it contains 13 critical cells (see figure 10-up): 1 0-cell, 6 1-cells and 6 2-cells. Let us comment that it is not the best DGVF possible. After the correction, there is only 1 critical 0-cell, which corresponds to the Betti numbers of the complex. Thus, there are two cycles in the Morse graph.

For the Dunce hat we used a simplicial complex from [24] consisting of 8 0-cells, 24 1-cells and 17 2-cells. The initial DGVF gives 3 critical cells (see figure 11-up): one of each dimension. After one step of the correction, we reverse a V-path and we finish, obtaining only 1 critical 0-cell, which is in accordance with the Betti numbers of the complex. The two cycles created in the homological DVF are shown in purple.
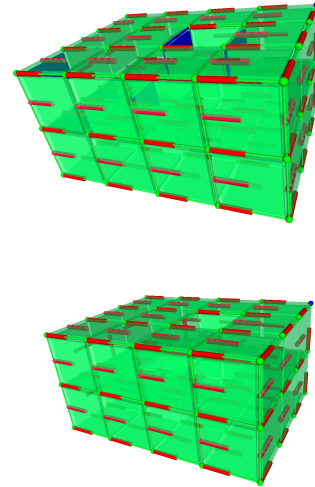

Fig. 10. Up: a DGVF over the Bing's house. Down: a perfect DVF obtained on the Bing's house. There is only one critical 0-cell (in blue)

VI. CONCLUSION AND FUTURE WORK

This approach not only gives a nice theoretical result (computing homological information with a gradient vector field on spaces where it is impossible if we restrict ourselves to the Discrete Morse Theory), but a promising computation tool. The worst-case complexity in time of one correction step is estimated to be $O(n^2) + O(N^3)$, where n is the number of voxels in the discrete object and N the number of confluence vertices. This is due to the system of linear equations systems, but in practice it works much faster. The number of correction steps depends on the quality of the initial DGVF.

In the immediate future, we have the intention of progressing in the following directions:

- We obtain our HDVF by correcting a DGVF. We would like to directly build it with cycles, without starting from an initial DGVF;

- Since confluence vertices force to solve systems of linear equations, we should try to minimize their number. This should be done when choosing a critical cell for cancelling. In our current implementation, this choice is completely arbitrary;

- In algorithm 4, after cancelling two critical cells we recompute the confluence vertices and their images by f. It does not seem difficult to restrict this computation. For example, if we cancelled two critical cells of dimension q and $q + 1$, then we could only recompute the confluence vertices of those dimensions.

REFERENCES

[1] F. Sergeraert, "Effective homology, a survey," http://www-fourier.ujf-grenoble.fr/ sergerar/Papers/Survey.pdf, 1992, [Online; accessed 11-June-2014].

[2] R. Forman, "A user's guide to discrete morse theory," *Seminaire Lotharingin de Combinatoire*, vol. 48, p. B48c, 2002. [Online]. Available: http://www.emis.ams.org/journals/SLC/wpapers/s48forman.pdf
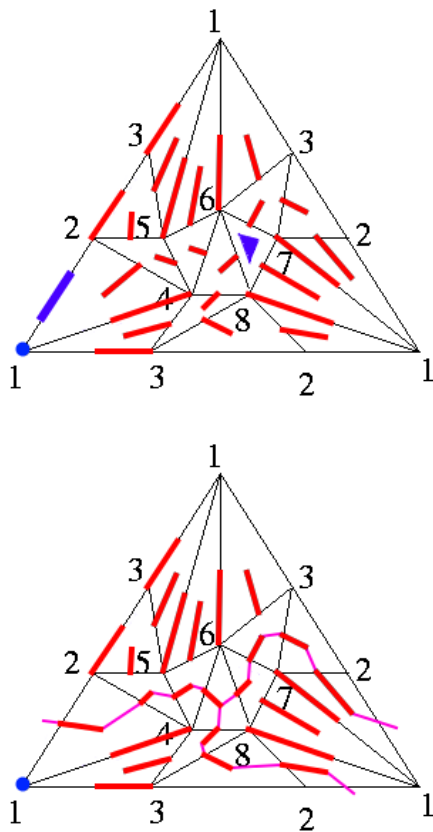
Fig. 11. Up: a DGVF over the Dunce hat with three critical cells in blue. Down: the perfect DVF given by the correction step. The only one critical cell is 1. The two cycles in the graph are displayed in purple.

graph pyramids," *Image and Vision Computing (IMAVIS)*, vol. 27, no. 7, pp. 846–853, June 2009. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2008.06.009

[13] C. J. A. Delfinado and H. Edelsbrunner, "An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere," *Computer Aided Geometric Design*, vol. 12, no. 7, pp. 771–784, 1995.

[14] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete and Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[15] D. Boltcheva, D. Canino, S. Merino Aceituno, J. C. Léon, L. De Floriani, and F. Hétroy, "An iterative algorithm for homology computation on simplicial shapes," *CAD Computer Aided Design*, vol. 43, no. 11, pp. 1457–1467, 2011.

[16] T. Lewiner, H. Lopes, and G. Tavares, "Toward optimality in discrete morse theory." *Experimental Mathematics*, vol. 12, no. 3, pp. 271–285, 2003. [Online]. Available: http://eudml.org/doc/52321

[17] A. Engstrom, "Discrete morse functions from fourier transforms," *Experimental Mathematics*, vol. 18, no. 1, pp. 45–54, 2009. [Online]. Available: http://projecteuclid.org/euclid.em/1243430528

[18] H. Molina-Abril and P. Real, "Homological optimality in discrete morse theory through chain homotopies," *Pattern Recognition Letters*, vol. 33, no. 11, pp. 1501 – 1506, 2012, computational Topology in Image Context.

[19] P. Dotko and H. Wagner, "Computing homology and persistent homology using iterated morse decomposition," *arXiv preprint arXiv:1210.1429*, pp. 1–31, 2012. [Online]. Available: http://arxiv.org/abs/1210.1429

[20] "DGtal: Digital geometry tools and algorithms library," http://dgtal.org.

[21] "Eigen," http://eigen.tuxfamily.org.

[22] "Graphviz," http://www.graphviz.org/.

[23] R. Ayala, D. Fernández-Ternero, and J. A. Vilches, "Perfect discrete morse functions on 2-complexes," *Pattern Recogn. Lett.*, vol. 33, no. 11, pp. 1495–1500, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2011.08.011

[24] M. Hachimori, "Simplicial complex library," http://infoshako.sk.tsukuba.ac.jp/ hachi/math/library/index_eng.html.

[3] H. Molina-Abril and P. Real, "Homological spanning forest framework for 2d image analysis," *Annals of Mathematics and Artificial Intelligence*, vol. 64, no. 4, pp. 385–409, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10472-012-9297-7

[4] J. R. Munkres, *Elements of algebraic topology.* Addison-Wesley, 1984.

[5] A. Hatcher, *Algebraic topology*. Cambridge, New York: Cambridge University Press, 2002.

[6] J.-L. Mari and P. Real, "Simplicialization of digital volumes in 26-adjacency: Application to topological analysis," *Pattern Recognition and Image Analysis*, vol. 19, no. 2, pp. 231–238, 2009.

[7] P. Real and H. Molina-Abril, "Cell at-models for digital volumes," in *Graph-Based Representations in Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Torsello, F. Escolano, and L. Brun, Eds. Springer Berlin Heidelberg, 2009, vol. 5534, pp. 314–323.

[8] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology.* Springer, 2004, vol. 157, no. 2, ch. 2, 7, pp. 255–258.

[9] M. Giesbrecht, "Probabilistic computation of the smith normal form of a sparse integer matrix," in *Algorithmic Number Theory*, ser. Lecture Notes in Computer Science, H. Cohen, Ed. Springer Berlin Heidelberg, 1996, vol. 1122, pp. 173–186.

[10] J.-G. Dumas, F. Heckenbach, D. Saunders, and V. Welker, "Computing simplicial homology based on efficient smith normal form algorithms," in *Algebra, Geometry and Software Systems*, M. Joswig and N. Takayama, Eds. Springer Berlin Heidelberg, 2003, pp. 177–206.

[11] T. Kaczyski, M. Mrozek, and M. lusarek, "Homology computation by reduction of chain complexes," *Computers and Mathematics with Applications*, vol. 35, no. 4, pp. 59–70, 1998.

[12] S. Peltier, A. Ion, W. Kropatsch, G. Damiand, and Y. Haxhimusa, "Directly computing the generators of image homology using