Homological Computation Using Spanning Trees[*]

H. Molina-Abril[1,2] and P. Real[1]

[1] Departamento de Matematica Aplicada I, Universidad de Sevilla,
{real,habril}@us.es
http://ma1.eii.us.es/
[2] Vienna University of Technology, Faculty of Informatics, PRIP Group
habril@prip.tuwien.ac.at
http://www.prip.tuwien.ac.at/

Abstract. We introduce here a new $\mathbb{F}_2$ homology computation algorithm based on a generalization of the spanning tree technique on a finite 3-dimensional cell complex K embedded in $\mathbb{R}^3$. We demonstrate that the complexity of this algorithm is linear in the number of cells. In fact, this process computes an algebraic map ϕ over K, called homology gradient vector field (HGVF), from which it is possible to infer in a straightforward manner homological information like Euler characteristic, relative homology groups, representative cycles for homology generators, topological skeletons, Reeb graphs, cohomology algebra, higher (co)homology operations, etc. This process can be generalized to others coefficients, including the integers, and to higher dimension.

Keywords: Cell complex, chain homotopy, digital volume, homology, gradient vector field, tree, spanning tree.

1 Introduction

Homology (providing a segmentation of an object in terms of its n-dimensional holes) is one of the pillar of Topological Pattern Recognition. To compute homology for a nD digital object (with $n \geq 3$) is cubic in time with regards to the number n of cells [9,2,8]. Classical homology algorithms reduce the problem to Smith diagonalization, where the best available algorithms have supercubical complexity [12]. An alternative to these solutions are the reduction methods. They iteratively reduce the input data by a smaller one with the same homology, and compute the homolgy when no more reductions are possible [8,10].

To have at hand an algorithm computing homology in $O(n)$ is one of the main challenge in this area and has been recently conjectured in [8].
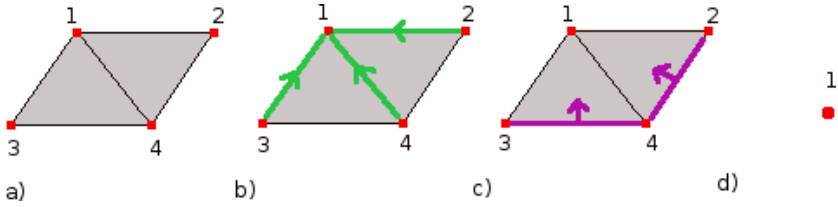
Fig. 1. (a) A cell complex K,(b) the first level of the forest determined by 0-cells and 1-cells, where $\phi(\langle 3 \rangle) = \langle 1, 3 \rangle$, $\phi(\langle 4 \rangle) = \langle 1, 4 \rangle$ and $\phi(\langle 2 \rangle) = \langle 1, 2 \rangle$, (c) the second level of the forest determined by 1-cells and 2-cells, where $\phi(\langle 3, 4 \rangle) = \langle 1, 3, 4 \rangle$ and $\phi(\langle 2, 4 \rangle) = \langle 1, 2, 4 \rangle$, (d) $H_0(K) = \langle 1 \rangle$

A finite cell complex K is a graded set formed of cells, with an operator ∂ describing the boundary of each cell in terms of linear combination of its faces. The finite linear combination (with coefficients in $\mathbb{F}_2 = \{0, 1\}$) of cells form a graded vector space called chain complex associated to three dimensional cell complex K embedded in $\mathbb{R}^3$ and denoted by $C_*(K; \mathbb{F}_2)$. In [6] the solution to the homology computation problem (calculating n-dimensional holes) of K is described in the following terms: to find a concrete linear map $\phi : C_*(K; \mathbb{F}_2) \to C_{*+1}(K; \mathbb{F}_2)$, increasing the dimension by one and satisfying that $\phi\phi = 0$ (nilpotency condition), $\phi\partial\phi = \phi$ (chain contraction condition) and $\partial\phi\partial = \partial$ (cycle condition). In [5], a map ϕ of this kind is called homology gradient vector field (HGVF). This datum ϕ is, in fact, a chain homotopy operator on K (a purely homological algebra notion) and it is immediate to establish a strong algebraic link between the cell complex associate to K and its homology groups ($H_0(K)$, $H_1(K)$, $H_2(K)$).

In [7] the homological deformation process ϕ is codified to a minimal homological expression in terms of mixed trees. Different strategies for building these trees give rise to useful results in segmentation, analysis, topological skeleton, multiresolution analysis, etc. But the complexity of this solution for the homology computation problem is still cubic.

In this paper, we follow a different approach which allows to reduce the complexity of the problem. In the incidence graph of the cell complex K (in which the cells are represented by points and the (non-oriented) edges are determined by the relation "to be in the boundary of"), we perform a sort of spanning tree technique. This process gives as output a three-level forest (the first level determined by 0 and 1-cells, the second one by 1 and 2 cells, the third one by 2 and 3-cells).

A theoretical result will guarantee that considering some conditions during the generation of this forest, it can be seen as a HGVF. In this way the process for getting the homology generators of K starting from ϕ is $O(n)$ in time, where n is the number of cells of K.

In Section 2, we will show that a spanning forest for a 1-dimensional finite cell complex K gives raise to an HGVF $\phi : C_*(K) \to C_{*-1}(K)$. In Section 3 this result is extended to 3-dimensional finite cell complexes.

2   Spanning Trees as a Homology Gradient Vector Fields

Before presenting this new approach, some notions about algebraic topology must be introduced. A q–*chain* a of a three-dimensional cell complex K is a formal sum of cells of $K^{(q)}$ ($q = 0, 1, 2, 3$). Let us consider the ground ring as the finite field $\mathbb{F}_2 = \{0, 1\}$. The q–chains form a group with respect to the component–wise addition; this group is the qth *chain complex* of K, denoted by $C_q(K)$. There is a chain group for every integer $q \geq 0$, but for a complex in $\mathbf{R}^3$, only the ones for $0 \leq q \leq 3$ may be non–trivial. The boundary map $\partial_q : C_q(K) \to C_{q-1}(K)$ applied to a q–cell σ gives us the collection of all its $(q-1)$–faces which is a $(q-1)$–chain. We say that $\sigma' \in \partial_q(\sigma)$ if σ' is a face of the q-cell σ. By linearity, the boundary operator ∂_q can be extended to q–chains, and satisfies $\partial_{q-1}\partial_q = 0$. From now on, a cell complex will be denoted by (K, ∂). A chain $a \in C_q(K)$ is called a q–*cycle* if $\partial_q(a) = 0$. If $a = \partial_{q+1}(a')$ for some $a' \in C_{q+1}(K)$ then a is called a q–*boundary*. Define the qth *homology group* to be the quotient group of q–cycles and q–boundaries, denoted by $H_q(K)$. For example in Figure 1, $\partial(\langle 2, 3, 4\rangle) = \langle 2, 3\rangle + \langle 2, 4\rangle + \langle 3, 4\rangle$, and the tree edges are faces of the 2-cell $\langle 2, 3, 4\rangle$. The 1-chain $\langle 2, 3\rangle + \langle 2, 4\rangle + \langle 3, 4\rangle$ is a 1–*cycle* and a 1–*boundary*.

Let (K, ∂) be a finite cell complex. A linear map of chains $\phi : C_*(K) \to C_{*+1}(K)$ is a *combinatorial gradient vector field* (or, shortly, combinatorial GVF) on K if the following conditions hold: (1) For any cell $a \in K_q$, $\phi(a)$ is a $q+1$-cell b; (2) $\phi^2 = 0$. Removing the first condition, ϕ will be called an *algebraic gradient vector field*. An algebraic GVF satisfying the conditions $\phi\partial\phi = \phi$ and $\partial\phi d = \partial$ will be called a *homology GVF* [6]. If ϕ is a combinatorial GVF which is only non-null for a unique cell $a \in K_q$ and satisfying the extra-condition $\phi\partial\phi = \phi$, then it is called a (combinatorial) *integral operator* [3]. An algebraic GVF ϕ is called *strongly nilpotent* if it satisfies the following property: Given any $u \in K^{(q)}$, if $\phi(u) = \sum_{i=1}^{r} v_i$ then $\phi(v_i) = 0$ for all $i = 1, \ldots, r$. We say that a linear map $f : C_*(K) \to C_*(K)$ is *strongly null over an algebraic gradient vector field* ϕ if given any $u \in K^{(q)}$, if $\phi(u) = \sum_{i=1}^{r} v_i$ then $f(v_i) = 0$ for all $i = 1, \ldots, r$.

Let (K, ∂) be a finite one dimensional cell complex (undirected graph) having only one connected component. The boundary operator $\partial : C_1(K) \to C_0(K)$ for a 1-cell e is given by $\partial(e) = v_2 + v_1$ (in $\mathbb{F}_2 = \{0, 1\}$), where v_1, v_2 are the endpoints of e. The boundary operator $\partial(w)$ for a 0-cell w is zero. Let $T = (V, E)$ a spanning tree (a tree composed of all the vertices) for K. Let us fix a root $v \in V$ for T and let us define the linear map $\phi : C_*(K) \to C_{*+1}(K)$ by

$\phi(w) = \{$the unique path from w to v through the tree $T\}$, $\forall w \in V$ and zero for every 1-cell of K.

In this definition, we understand by path a sum of edges in T connecting w with the root v. Then, the composition $\phi\phi$ is obviously zero and the conditions $\phi\partial\phi = \phi$ and $\partial\phi\partial = \partial$, where ∂ is the boundary operator for K, are also satisfied for every cell of K. In consequence:

Proposition 1. *The map ϕ described above determine a HGVF for the 1-dimensional cell complex K.*
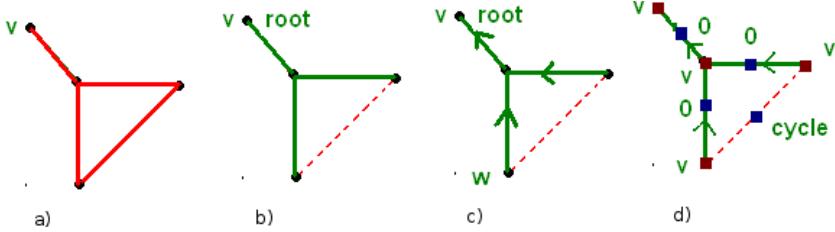
Fig. 2. (a) A graph K,(b) a spanning tree T, (c) description of ϕ, (d) description of π

Let $\pi : C_*(K) \to C_*(K)$ be a linear map defined by $\pi = id_{\mathcal{C}(\mathcal{K})} - \partial\phi - \phi\partial$. For each 0-cell w of K, $\pi(w) = v$. For each 1-cell e of K, $\pi(e) = 0$ if e belongs to T, and $\pi(e) = \{$ a representative cycle of a homology generator for $K\}$ if e does not belong to T. Let us consider now the incidence graph $IG(K)$ for K, that is, a graph with one vertex per point (red vertices forming the set V_r), one vertex per edge of K (blue vertices forming the set V_b) and an edge for every incidence between a point of V and a line belonging to E (see Figure 2). In other words, $IG(K)$ is the Hasse diagram [1] of the set of cells partially ordered by the relation of "to be a face of".The map π can be described as a function $\pi : V_r \bigcup V_b \to \operatorname{Ker} \partial$, which provides representative cycles of the different homology generators of K (evaluating π for those blue vertices not in T).

3   Homology Computation in Linear Time

Throughout this section, the extension of the previous spanning tree technique to higher dimensions is presented. A linear time algorithm for homology computation with coefficients in $\mathbb{F}_2$ is given.

Let (K, ∂) be a finite three-dimensional cell complex. Without loss of generality, suppose that K has only one connected component. Let consider the incidence graph $IG(K) = (V, E)$ for K, defined by the graph with one vertex per cell, and one edge for each incidence between an i-cell and an $i + 1$-cell. The set of vertices and edges for $IG(V)$ can be decomposed in the following way:

$$V = \bigcup_{i=0}^{3} \{i\text{-cells for } K\}$$

$$E = \bigcup_{i=1}^{3} \{\text{unordered pairs } \{\sigma', \sigma\}, \text{ where } \sigma' \in \partial_i\sigma\}$$

Let $T^0 = (V^0, E^0)$ a tree spanning the vertices $V_0 = K_0$ in $IG(K)$. In T^0, $V^0 = V_0^0 \cup V_1^0$ (that we called, respectively, red and blue vertices of T^0), with $V_0^0 = V_0$, $V_1^0 \subset K_1$. Let us fix a red vertex ($v_0 \in K_0$) as the root of the tree T^0. Starting from the root v_0, let us obtain the maximum number of pairwise distints arcs whose tail is a red vertex and its head is a blue vertex. For doing this, we simply generate those arcs in T^0 from the edges composing the branches,
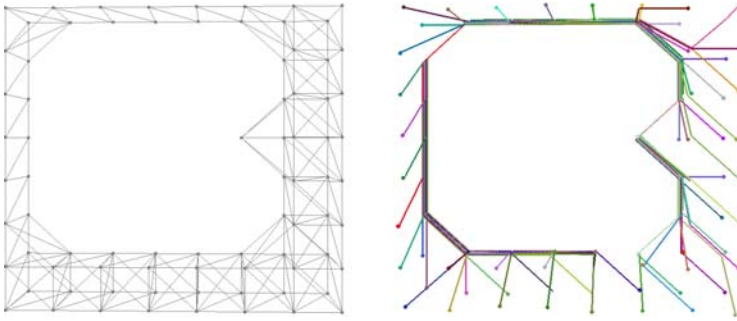
Fig. 3. (a)A 3-dimensional cell complex (b) T_0 tree

with tail being a red vertex and pointing them towards the root. Let us define $\phi_0(w)$ for a vertex w in K_0 by the sum of all the edges forming the unique path joining w with the root v_0 ($\phi_0(v_0)$ will be 0). It is straightforward to verify that $\phi_0\partial_1\phi_0 = \phi_0$ and $\partial_1\phi_0\partial_1 = \partial_1$. An example of the calculation of T^0 over a real $3D$-image is shown in Figure 3.

Now, we calculate a forest $\mathcal{F}^1 = (V^1, E^1)$ in $IG(K)$ spanning the vertices $V_0^1 = K_1 \setminus V_1^0$. In this graph, $V^1 = V_0^1 \cup V_1^1$ (red and blue vertices of $\mathcal{F}^1$), with $V_1^1 \subset K_2$ and it is constructed with the conditions: (a) given $e \in V_0^1$, all the 2-cell c, vertices of $IG(K)$ having as part of its boundary to e must be in V_1^1 as well the edge connecting c to e; (b) that if a 2-cell c is in V_1^1, then all the edges in $IG(K) \setminus E^0$ specifying those edges in K that are in the boundary of c, must be in $\mathcal{F}^1$ (see Figure 4).

Let us fix a red vertex $v_1^j \in V_0^1$ ($j = 1, \ldots, k$) as a root for each of the trees $T_1^1, \ldots, T_m^1$ composing the forest $\mathcal{F}^1$. We only handle one of these tree T_1^1 and we do analogously for the others trees of $\mathcal{F}^1$. We first determine the red vertices in T_1^1 (1-cells in K) with degree greater or equal than three (they are called bifurcation red vertices). Among the 2-cells in V_0^1 having as a part of the boundary the red bifurcation vertex e, there will be a blue vertex $c_1(e)$ which is the parent of e and at least two more blue vertices $c_2(e), \ldots, c_r(e)$ which are the children of e in T_1^1. If v is a vertex of T_1^1, let us denote by $T_1^1(v)$ the subtree of T_1^1 generated by the descendants of v and their relationships in T_1^1 (including v as the root of $T_1^1(v)$). There is a semi-direct path from the red vertex w to the red vertex w' in T_1^1 if there is a sequence $w = w_1, z_1, \ldots, z_t, w' = w_{t+1}$ in which w_i are red vertices of T_1^1, z_i are blue vertices of T_1^1, (w_i, z_i) are arcs and $\{z_i, w_{i+1}\}$ are edges of T_1^1, for all $i = 1, \ldots, t$.

We now generate the maximum number of arcs (from the edges composing the branches) whose tail is a red vertex and pointing them away from the root. From this set, we eliminate those arcs (that is, we eliminate the arrow in the corresponding edge) that are associated to $n-1$ sons of a red bifurcation vertex of degree n. Let us define the map $\phi_1 : C_1(K) \to C_2(K)$. If $w \in V_1^0$, then $\phi_1(w) = 0$. If $w \in V_0^1$, then $\phi(w)$ is the sum of 2-cells (blue vertices in T_1^1) forming the different semi-directed pathes existing from w in $T_1^1(w)$.
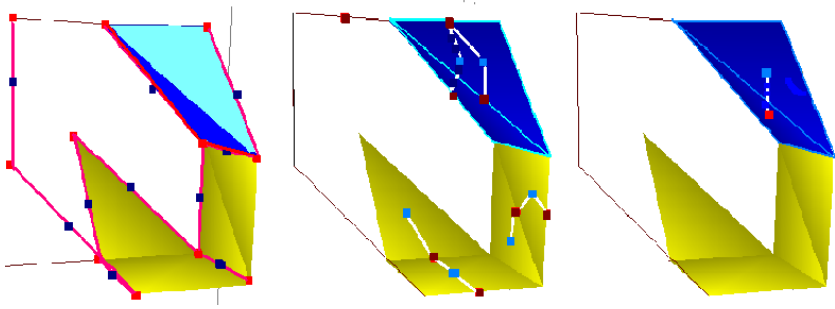
Fig. 4. (a) Spanning trees over the incidence graph of a 3-dimensional cell complex

This linear map verifies that $\phi_1\phi_0 = 0$, $\phi_1\partial_2\phi_1 = \phi_2$ and $\partial_2\phi_1\partial_2 = \partial_2$. The set of vertices $V_1^1 = \{e \in K_2 \ / \ e \in \phi_1(v), \text{ for some } v \in V_0^1\}$.

We now calculate a forest $\mathcal{F}^2 = (V^2, E^2)$ in $IG(K)$ spanning the vertices $V_0^2 = K_1 \setminus V_1^1$. In this graph, $V^2 = V_0^2 \cup V_1^2$ (red and blue vertices of $\mathcal{F}^2$), with $V_2^1 \subset K_3$ and it is constructed with the conditions: (a) given $e \in V_0^2$, all the 3-cell c, vertices of $IG(K)$, that have as part of its boundary to e must be in V_2^1 as well the edge connecting c to e; (b) that if a 3-cell c is in V_1^2, then all the edges in $IG(K) \setminus E^1$ specifying those 2-cells in K that are in the boundary of c, must be in $\mathcal{F}^2$ (see Figure 4). Using this forest, we define in an analogous way to $\mathcal{F}_1^1$, the map $\phi_2 : C_2(K) \to C_3(K)$. The set of vertices V_1^2 will agree with the set $\{e \in K_3 \ / \ e \in \phi_2(v), \text{ for some } v \in V_0^2\} \subset K_3$. Finally, ϕ_2 applied over an element of $K_3 \setminus V_1^2$ is zero.

The final map $\phi : C_*(K) \to C_{*+1}(K)$ satisfies the nilpotency, chain contraction and cycle conditions. The map $id_{\mathcal{C}(K)} + \partial\phi + \phi\partial$ applied to every leave of the corresponding tree provides us the different representative cycles for all the homology generators of K.

This process of $\mathbb{F}_2$-homology computation over a 3-dimensional cell complex, can be seen as the simple construction of three spanning trees but taking into account some special conditions. Considering a classical spanning tree technique as for example Depth-first search [13], which time complexity is $O(V + E)$ (V is the number of vertices of the graph and E the number of edges), the linearity of our method can be directly deduced.

4 Conclusions and Future Work

Many issues in computer imagery are related to the computation of homological information, like classification ([4] [11]), shape and pattern recognition ([10] [14]), etc. Image data require a huge amount of computational resources, and to find efficient algorithms which analyze image data is an active field of research. When dealing with 3–dimensional images, a fast computation is crucial, and it is even more with higher dimensionsal data.

A linear in time algorithm for computing homological information over a 3–dimensional cell complex is presented here. This method is based in spanning

tree strategies. The main advantage of this result is its low computational time cost, in comparison with the complexity of the existing cubic in time methods.

There exist several spanning tree strategies. Some of them run in logarithmic time by using parallelization. Due to this fact, as future work, we plan to apply this parallelized methods to the construction of the homological forest in order to increase efficiency.

Another future aims is to deal with integer homology, instead of restricting the coefficients to $\mathbb{F}_2$, and to apply this method to different structures.

References

1. Birkhoff, G.: Lattice Theory. American Mathematical Society, Providence (1948)
2. Delfinado, C.J.A., Edelsbrunner, H.: An incremental algorithm for betti numbers of simplicial complexes on the 3–sphere. Comput. Aided Geom. Design 12, 771–784 (1995)
3. González-Diaz, R., Jiménez, M.J., Medrano, B., Molina-Abril, H., Real, P.: Integral operators for computing homology generators at any dimension. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 356–363. Springer, Heidelberg (2008)
4. Madjid, A., Corriveau, D., Ziou, D.: Morse homology descriptor for shape characterization. In: ICPR 2004: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), Washington, DC, USA, vol. 4, pp. 27–30. IEEE Computer Society, Los Alamitos (2004)
5. Molina-Abril, H., Real, P.: Advanced homological information on 3d digital volumes. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 361–371. Springer, Heidelberg (2008)
6. Molina-Abril, H., Real, P.: Cell at-models for digital volumes. In: 7th IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition, Venice (Italy). LNCS, Springer, Heidelberg (2009)
7. Molina-Abril, H., Real, P.: Homological tree-based strategies for image analysis. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 326–333. Springer, Heidelberg (2009)
8. Mrozek, M., Pilarczykand, P., Zelazna, N.: Homology algorithm based on acyclic subspace. Computers and Mathematics with Applications 55, 2395–2412 (2008)
9. Munkres, J.: Elements of Algebraic Topology. Addison Wesley Co., Reading (1984)
10. Peltier, S., Ion, A., Kropatsch, W.G., Damiand, G., Haxhimusa, Y.: Directly computing the generators of image homology using graph pyramids. Image Vision Comput. 27(7), 846–853 (2009)
11. Scopigno, R., Zorin, D., Carlsson, G., Zomorodian, A., Collins, A., Guibas, L.: Persistence barcodes for shapes (2004)
12. Storjohann, A.: Near optimal algorithms for computing smith normal forms of integer matrices, pp. 267–274 (1996)
13. Tarjan, R.: Finding dominators in directed graphs. SIAM J. on Comput. 3, 62–89 (1974)
14. Zelawski, M.: Pattern recognition based on homology theory. MG&V 14(3), 309–324 (2005)