



Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavisDirectly computing the generators of image homology using graph pyramids[☆]Samuel Peltier^{a,b,*}, Adrian Ion^a, Walter G. Kropatsch^a, Guillaume Damiand^b, Yll Haxhimusa^{c,a}^aPattern Recognition and Image Processing Group, Faculty of Informatics, Vienna University of Technology, A-1040 Vienna, Austria^bUniversité de Lyon, CNRS, LIRIS, UMR5205, F-69622, France^cDepartment of Psychological Sciences, Purdue University, West Lafayette, IN 47907-2081, USA

ARTICLE INFO

Article history:

Received 14 October 2007

Received in revised form 30 March 2008

Accepted 29 June 2008

Keywords:

Homology generators

Irregular graph pyramids

ABSTRACT

We introduce a method for computing homology groups and their generators of a 2D image, using a hierarchical structure, i.e. irregular graph pyramid. Starting from an image, a hierarchy of the image is built by two operations that preserve homology of each region. Instead of computing homology generators in the base where the number of entities (cells) is large, we first reduce the number of cells by a graph pyramid. Then homology generators are computed efficiently on the top level of the pyramid, since the number of cells is small. A top down process is then used to deduce homology generators in any level of the pyramid, including the base level, i.e. the initial image. The produced generators fit on the object boundaries. A unique set of generators called the minimal set, is defined and its computation is discussed. We show that the new method produces valid homology generators and present some experimental results.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Handling 'structured geometric objects' is important for many applications related to geometric modeling, computational geometry, image analysis, etc. One has often to distinguish between different parts of an object, according to properties which are relevant for the application. For image analysis, a region is a (structured) set of pixels or voxels, or more generally a (structured) set of lower-level regions. At the lowest level of abstraction, such an object is a subdivision,¹ i.e. a partition of the object into cells of dimension 0, 1, 2, 3, ... (i.e. vertices, edges, faces, volumes, etc.) [1,2]. In general, combinatorial structures (graphs, combinatorial maps, n -G-maps, etc.) are used to describe objects subdivided into cells of different dimensions. The structure of the object is related to the decomposition of the object into sub-objects and to the relations between these sub-objects: basically, topological information is related to the cells and their adjacency or incidence relations. Further information (embedding information) is associated to these sub-objects and describes for instance their shapes (e.g. a point, respectively a curve, a part of a surface, is associated with each vertex, respectively each edge, each face), their textures or colors, or other information depending on the application. A common problem is to characterize

structural (topological) properties of handled objects. Different topological invariants have been proposed, like Euler characteristics, orientability, homology, etc. (see [3]).

Homology is a powerful topological invariant, which characterizes an object by its " p -dimensional holes". Intuitively the 0-dimensional holes can be seen as connected components, 1-dimensional holes can be seen as tunnels and 2-dimensional holes as cavities. For example, the torus in Fig. 1(a) contains one 0-dimensional hole, two 1-dimensional holes (each of them are an edge cycle) and one 2-dimensional hole (the cavity enclosed by the entire surface of the torus). This notion of p -dimensional hole is defined in any dimension. Another important property of homology is that local calculations induce global properties. In other words, homology is a tool to study spaces and has been applied in image processing for 2D and 3D image analysis [4,5]. Although in this paper, we use 2D binary images to show the proof of concept, we do not encourage usage of homology groups and generators to find connected components in 2D images, since efficient approaches already exist [6]. However, these 'classical' approaches cannot be easily extended for many problems that exist in higher dimensions, since our visual intuition is inappropriate and topological reasoning becomes important. Computational topology has been used in metallurgy [7] to analyze 3D spatial structure of metals in an alloy and in medical image processing [8] in analyzing blood vessels. In higher dimensional problems (e.g. beating heart represented in 4D) the importance of homology groups and generators becomes clear because of the nice and clean formulation which holds in any dimension (number of connected components, tunnels, holes, etc.). One can think of other applications, as a preprocessing step, to speed up recognition of complex shapes in large image

[☆] Partially supported by the Austrian Science Fund under Grants P18716-N13 and S9103-N04, and the USA Air Force Office of Scientific Research.

* Corresponding author. Address: Pattern Recognition and Image Processing Group, Faculty of Informatics, Vienna University of Technology, A-1040 Vienna, Austria. Tel.: +33 5 49 49 74 80.

E-mail addresses: peltier@sic.univ-poitiers.fr (S. Peltier), ion@prip.tuwien.ac.at (A. Ion), krw@prip.tuwien.ac.at (W.G. Kropatsch), guillaume.damiand@liris.cnrs.fr (G. Damiand), yll@psych.purdue.edu (Y. Haxhimusa).

¹ For instance, a Voronoi diagram in the plane defines a subdivision of the plane.

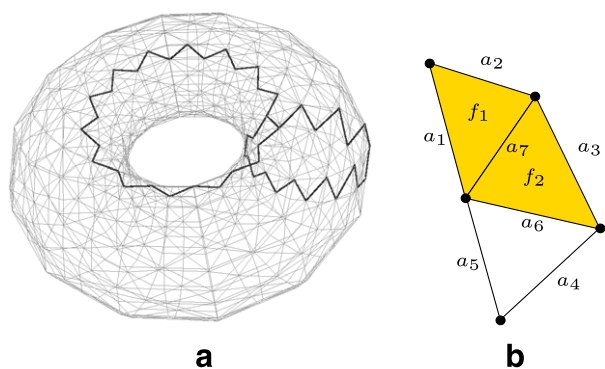


Fig. 1. (a) A triangulation of the torus. (b) A simplicial complex made of 1 connected component and containing one 1-dimensional hole.

databases, e.g. images are first filtered based on their topological invariants and afterward are matched using shapes, appearances, etc.

The usage of homology groups and generators in image processing is a new topic and is not widely spread. In this paper, we present a new method for computing homology groups and their generators using a hierarchical structure. This work was introduced in [9] and is build by using two operations: contraction and removal. These two operations are used also in [10] to incrementally compute homology groups and their generators of 2D closed surfaces, but a hierarchy is not build. In this paper, the complexity of the method is detailed, which shows the interest of our approach compared to Agoston's classical method. The current paper also considers the geometry of the produced generators, and defines and studies the concept of minimal generator set.

The paper is structured as follows. Basic notions on homology and irregular graph pyramids are recalled in Sections 2 and 3. The proposed method to compute homology groups and their generators is presented in Section 4, the complexity is given and the property of the resulting generators are detailed. Section 5 presents the concept of minimal generator set. Experimental results on 2D images that compare the result of the new method with Agoston's one are found in Section 6.

2. Homology

In this part, the basic homology notions of chain, cycle, boundary and homology generator are recalled. Interested readers can find more details in [11].

The homology of a subdivided object X can be defined in an algebraic way by studying incidence relations of its subdivision. Within this context, a cell of dimension p is called a p -cell and the notion of p -chain is defined as a sum $\sum_{i=1}^{nb\ p\text{-cells}} \alpha_i c_i$, where c_i are p -cells of X and α_i are coefficients assigned to each cell in the chain. Homology can be computed using any group \mathfrak{A} for the coefficients α_i . But, the theorem of universal coefficients [11] ensures that all homological information can be obtained by choosing $\mathfrak{A} = \mathbb{Z}$. It is also known [11] that for n D objects embedded in \mathbb{R}^n , homology information can be computed by simply considering chains with moduli 2 coefficients ($\mathfrak{A} = \mathbb{Z}/2\mathbb{Z}$). Note that in this case, a cell that appears twice on a chain vanishes, because $c + c = 0$ for any cell c when using moduli 2 coefficients (i.e. if a cell appears even times we discard it otherwise we keep it). In the following, only chains with coefficients over $\mathbb{Z}/2\mathbb{Z}$ will be considered. Note that the notion of chain is purely formal and the cells that compose a chain do not have to satisfy any property. For example, on the simplicial complex illustrated on Fig. 1(b) the sums: $a_1 + a_4$, a_3 and $a_2 + a_7 + a_4$ are 1-chains.

For each dimension $p = 0, \dots, n$, where $n = \dim(X)$, the set of p -chains forms an abelian group denoted C_p . The p -chain groups can be put into a sequence, related by applications ∂_p describing the boundary of p -chains as $(p - 1)$ -chains:

$$C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

which satisfy $\partial_p \partial_{p-1}(c) = 0$ for any p -chain c , i.e. the boundary of a cell is a null chain. This sequence of groups is called a *free chain complex*.

The boundary of a p -chain reduced to a single cell is defined as the sum of its incident $(p - 1)$ -cells. The boundary of a general p -chain is then defined by linearity as the sum of the boundaries of each cell that appears in the chain, e.g. in Fig. 1(b), $\partial(f_1 + f_2) = \partial(f_1) + \partial(f_2) = (a_1 + a_2 + a_7) + (a_7 + a_3 + a_6) = a_1 + a_2 + a_3 + a_6$. Note that as mentioned before, chains are considered over $\mathbb{Z}/2\mathbb{Z}$ coefficients, i.e. any cell that appears twice vanishes.

For each dimension p , the set of p -chains which have a null boundary are called p -cycles and are a subgroup of C_p , denoted Z_p , e.g. $a_1 + a_2 + a_7$ and $a_7 + a_3 + a_6 + a_3$ are 1-cycles. The set of p -chains which bounds a $p + 1$ -chain are called p -boundaries and they are a subgroup of C_p , denoted B_p , e.g. $a_1 + a_2 + a_7 = \partial(f_1)$ and $a_1 + a_6 + a_3 + a_2 = \partial(f_1 + f_2)$ are 1-boundaries.

According to the definition of a free chain complex, the boundary of a boundary is the null chain. Hence, this implies that any boundary is a cycle. Note that according to the definition of a free chain complex, any 0-chain has a null boundary, hence every 0-chain is a cycle.

The p th homology group, denoted H_p , is defined as the quotient group Z_p/B_p . Thus, elements of the homology groups H_p are equivalence classes and two cycles z_1 and z_2 belong to the same equivalence class if their difference is a boundary (i.e. $z_1 = z_2 + b$, where b is a boundary). Such two cycles are called *homologous*, e.g. let $z_1 = a_5 + a_4 + a_3 + a_7$, $z_2 = a_5 + a_4 + a_6$ and $z_3 = a_1 + a_2 + a_3 + a_6$; z_1 and z_2 are homologous ($z_1 = z_2 + \partial(f_2)$) but z_1 and z_2 are not homologous to z_3 . Let H_p be a homology group generated by q independent equivalence classes $\mathcal{C}_1, \dots, \mathcal{C}_q$, any set $\{h_1, \dots, h_q \mid h_1 \in \mathcal{C}_1, \dots, h_q \in \mathcal{C}_q\}$ is called a *set of generators* for H_p . For example, either $\{z_1\}$ or $\{z_2\}$ can be chosen as a generator of H_1 for the object represented in Fig. 1(b).

Note that some of the notions mentioned before could be confused with similar notions from graph theory. Table 1 associates these homology notions with notions classically used in graph theory. Note that a closed path of faces is a closed surface homeomorphic to a sphere.

3. Irregular graph pyramids

In this part, basic notions of pyramids, like receptive field, contraction kernel and equivalent contraction kernel, are introduced. For more details see [12].

A pyramid (Fig. 2a) describes the contents of an image at multiple levels of resolution. A high resolution input image is at the base level. Successive levels reduce the size of the data by a *reduction factor* $\lambda > 1.0$. The *reduction window* relates one cell at the reduced level with a set of cells in the level directly below. The contents of a lower resolution cell are computed by means of a *reduction function*

Table 1
Translation of homology notions to graph theory

Homology theory	Graph theory
0-Cell, 1-cell, 2-cell	Vertex, edge, face
0-Chain, 1-chain, 2-chain	Set of vertices, set of edges, set of faces
0-Cycle, 1-cycle, 2-cycle	Set of vertices, closed path of edges, closed path of faces

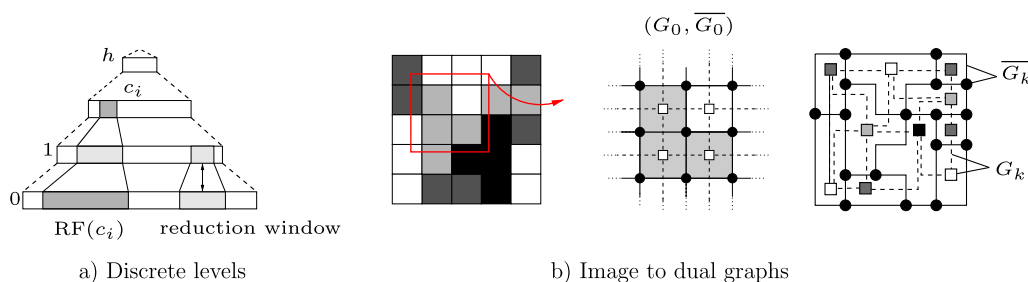


Fig. 2. (a) Pyramid concept, and (b) representation of the cells and their neighborhood relations by a pair of dual plane graphs at the level 0 and k of the pyramid.

the input of which are the descriptions of the cells in the reduction window. Two successive levels of a pyramid are related by the reduction window and the reduction factor. Higher level descriptions should be related to the original input data in the base of the pyramid. This is done by the *receptive field* (RF) of a given pyramidal cell c_i . The RF (c_i) aggregates all cells (pixels) in the base level of which c_i is the ancestor.

Each level represents a partition of the pixel set into cells, i.e. *connected subsets of pixels*. The construction of an irregular pyramid is iteratively local [13]. On the base level (level 0) of an irregular pyramid the cells represent single pixels. A cell on level $k + 1$ (parent) is a union of neighboring cells on level k (children). This union is controlled by so called *contraction kernels* (CK) [14], a spanning forest which relates two successive levels of a pyramid. Every parent computes its values independently of other cells on the same level. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid. Neighborhoods on level $k + 1$ are derived from neighborhoods on level k . Higher level descriptions are related to the original input by the *equivalent contraction kernels* (ECK). A level of the graph pyramid consists of a pair (G_k, \overline{G}_k) of plane graphs G_k and its geometric dual \overline{G}_k (Fig. 2b). The planarity of graphs restricts us to use only the 4-connectivity of the pixels. The vertices of G_k represent the cells on level k and the edges of G_k represent the neighborhood relations of the cells, depicted with square vertices and dashed edges in Fig. 2b. The edges of \overline{G}_k represent the borders of the cells on level k , solid lines in Fig. 2b, including so called pseudo edges needed to represent neighborhood relations to a cell completely enclosed by another cell. Finally, the vertices of \overline{G}_k (circles in Fig. 2b), represent junctions of border segments of \overline{G}_k . The sequence (G_k, \overline{G}_k) , $0 \leq k \leq h$ is called irregular (dual) graph pyramid. For simplicity of the presentation the dual \overline{G} is omitted afterward.

4. Computing homology generators in a graph pyramid

There exists a general method for computing homology groups. This method is based on the transformation of incidence matrices [11] (which describe the boundary homomorphisms) into their reduced form called *Smith normal form*. Agoston proposes a general algorithm, based on the use of a slightly modified Smith normal form, for computing a set of generators of these groups [3]. Even if Agoston's algorithm is defined in any dimension, the main drawback of this method is directly linked to the complexity of the reduction of an incidence matrix into its Smith normal form, which is known to consume a huge amount of time and space. Another well known problem is the possible appearance of huge integers during the reduction of the matrix. A more complete discussion about Smith normal algorithm complexity can be found in [15]. Indeed, Agoston's algorithm cannot directly be used for computing homology generators and different kinds of optimisations have been proposed.

Based on the work of [16,17], an optimisation for the computation of homology generators, based on the use of sparse matrices and moduli operations has been proposed [18]. This method avoids the possible appearance of huge integers. The authors also observed an improvement of time complexity dropping from $O(n^2)$ to $O(n^{5/3})$, where n is the number of cells of the subdivision.

An algorithm for computing the rank of homology groups, i.e. the Betti numbers has been proposed in [19]. The main idea of this algorithm is to reduce the number of cells of the initial object in order to obtain a homologically equivalent object, made out of less cells. In some special cases (orientable objects), Betti numbers can directly be deduced from the resulting object. However, this method cannot directly provide a set of generators. Based on the previously mentioned work, an algorithm for computing a minimal representation of the boundary of a 3D voxel region, from which homology generators can directly be deduced has been defined in [10].

4.1. Description of the new method

The method we propose in this paper has the same philosophy as the methods of Kaczynski and Damiand [20,21]: reducing the number of cells of an object for computing homology. Moreover, we keep all simplifications that are computed during the reduction process by using a pyramid. In this way, homology generators can be computed in the top level of the pyramid, and can be used to deduce generators of any level of the pyramid. In particular, we show how generators of the higher level can be directly down-projected on the desired level (using equivalent contraction kernels).

Starting from an initial image, we build an irregular graph pyramid. We show in Section 4.2 that building pyramids by decimation using contraction kernels [14] preserves homology of a subdivided object. Indeed, homology of the initial image can be computed in any level of the pyramid, and in particular in the top level where the object is described with the smallest number of cells.

Moreover, we use the notions of receptive field and equivalent contraction kernel of a vertex and the notion of surviving edge (see Sections 3 and 4.3), and show that the generators of homology groups of any level of the pyramid can be deduced from those computed on the highest level. Note that in special cases, the highest level of the pyramid may be reduced to exactly a set of generators of the initial image, as shown in [10].

Our method can be summarized by the following steps:

- (1) Starting from a labeled image, a graph pyramid $\{G_0, G_1, \dots, G_k\}$ is built using contraction kernels of vertices of the primal, with the same label.
- (2) Homology groups generators are computed for \overline{G}_k .
- (3) Homology generators of any level i can be deduced from those of level $i + 1$ using the contraction kernels and surviving edges of the dual. In particular, we obtain the homology generators of the initial image.

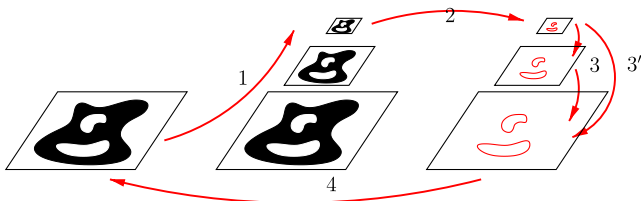


Fig. 3. Computing generators of homology groups using an image pyramid.

Note that homology generators of the lowest level can directly be deduced from the highest level using the notion of equivalent contraction kernel (arrow 3' in Fig. 3). Fig. 3 illustrates the general method that we propose for computing homology generators of an image.

4.2. Preserving homology on irregular graph pyramids

The algorithm described in [19] is based on operations of *interior face reduction* that reduce the number of cells of the subdivision. The main idea is to find a p -cell x and a $(p + 1)$ -cell y , such that x is incident to y . Then x and y are removed and the boundary of the neighboring p -cells that were adjacent to x are modified such that the new boundary $\partial(z)$ of cell z is defined as its initial boundary added with the boundary of y . Indeed, if x is incident to exactly two p -cells y and y' , the result of the corresponding interior face reduction can be seen as the removal of x and the merging of y and y' . It is proved in [19] that interior face reduction preserves homology.

Observing the dual graph, the operations of contraction and removal that are used to build each level of the pyramid are interior face reduction: two faces that are merged share a common edge that is removed, and an edge is contracted if one of its endpoints is incident to exactly two different edges, i.e. if it is incident to a dual vertex of vertex degree less than 3. Thus, homology is preserved in every level of the pyramid.

4.3. Delineating generators

A 1D generator in $\bar{G}_k = (\bar{V}_k, \bar{E}_k)$ is a closed path connecting vertices of \bar{G}_k and surrounding at least one hole. Each vertex $\bar{v} \in \bar{G}_k$ is the result of contracting a tree (contraction kernel \bar{CK}) of \bar{G}_{k-1} . Each edge $(\bar{v}_1, \bar{v}_2) \in \bar{G}_k$ corresponds to a surviving edge $(\bar{w}_1, \bar{w}_2) \in \bar{G}_{k-1}$ with $\bar{w}_1 \in \bar{CK}_{k-1}(\bar{v}_1)$ and $\bar{w}_2 \in \bar{CK}_{k-1}(\bar{v}_2)$, i.e. an edge that has neither been contracted nor removed.²

Given a generator in \bar{G}_k , mapping it to the level below is done by identifying the surviving edges in \bar{G}_{k-1} corresponding to the generator edges in \bar{G}_k and, where the generator is disconnected, adding paths to fill in the gaps and reconnect. For every two consecutive edges not having a common vertex in \bar{G}_{k-1} but having one in \bar{G}_k , the unique path connecting their disconnected endpoints in the contraction kernel $\bar{CK} \subset \bar{G}_{k-1}$ of their shared vertex in \bar{G}_k is added.

Because each path added in \bar{G}_{k-1} is entirely part of a contraction kernel, with contraction being used in the dual only for boundary simplification purposes, never connecting two different boundaries, and because the building process preserves homology (see Section 4.2) the obtained generators will be homologous to the ones in \bar{G}_k .

Reiterating this process of mapping the generator cycles of \bar{G}_k from k to $k - 1, \dots$ to 0, cycles in \bar{G}_0 corresponding to the generators of the top level can be identified. By replacing the contraction kernels, with the equivalent contraction kernels, using the same methodology, the generator cycles of \bar{G}_k can be directly mapped to \bar{G}_0 . For an example, see Fig. 4.

² Not part of any simplification.

4.4. Complexity of the method

The global complexity of our method is the sum of the complexity of the pyramid construction, Agoston's method (on the top level of the pyramid) and the down-projected (on level i).

- (1) The complexity of the pyramid construction is in $O(p \log(p))$ with p the number of pixels.
- (2) The homology group generators are computed by using Agoston's method in $O(n_k^2)$ with n_k the number of cells of the top level of the pyramid.
- (3) the down-projection of generators of level i from those of level $i + 1$ is achieved in $O(n_{i+1})$ with n_{i+1} the number of cells of level $i + 1$. As shown in Section 4.3, we can compute directly generators of level i by down-projecting generators of level k . Thus the complexity is in $O(n_i)$ with n_i the number of cells of level i .

The global complexity of our method for level i is thus in $O(p \log(p) + n_k^2 + n_i)$. By comparison, the classical Agoston method is in $O(n_i^2)$.

We can conclude that the gain of our method increases as the difference between n_k and n_i increases. Moreover, as shown in experiments (see Section 6), in practical cases, n_k is really smaller than n_i and the gain is important.

4.5. Controlling the geometry of the generators

When computing homology generators with Agoston's method, directly on the initial image, we cannot have any control of their geometry. More precisely, the aspect of the obtained generators is directly linked to the construction of incidence matrices, which is determined by the scanning of each cell of the initial image (see [18] for a first study of the influence of different parameters on the geometry of generators).

We prove in this section that for 2D images, the presented method provides a set of generators that always fit on the borders of a region R . In the following, an edge on the border of a 2D region is called a border edge.

First, we show that any 1-cycle in the top level of the pyramid contains only border edges. Second, we show that the down-projection of a 1-cycle composed of border edges, is still a cycle composed of border edges.

Property 1. Any 1-cycle in the top level of the pyramid computed with the presented method contains only border edges.

Proof. On the top level, a region is represented by a unique 2D-cell. Hence each edge of the top level is either a border edge or an edge linking two different borders of R (we call it a pseudo edge).

Let z be a 1-cycle on the top level, if z contains any pseudo edge $e = (v_1, v_2)$, where v_1 and v_2 are two vertices that stand on two different borders b_1 and b_2 of R , then in order for z to be a 1-cycle there exists another edge $e' = (v'_1, v'_2) \neq e$ between b_2 and b_1 . Note that $e \neq e'$ because $e - e'$ is a null 1-chain, and in such a case, e would not belong to z . Therefore edges e and e' divide R into at least two 2D-cells, which is not possible as any region on the top level is made of only one cell. Hence, any 1-cycle on the top level of the pyramid contains only border edges. \square

Let us consider Fig. 5(b), which represents the top level of the pyramid built from the initial image represented in Fig. 5(a). The subdivision is made of one 2D-cell R_1 ; four border edges e_1, e_2, e_3, e_4 ; two pseudo edges e_5, e_6 ; and four vertices. Property 1 ensures that for this subdivision, any 1-cycle can be written as $\alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 + \alpha_4 e_4$, where $\alpha_i = 0, 1, i = 1 \dots 4$. Note that for

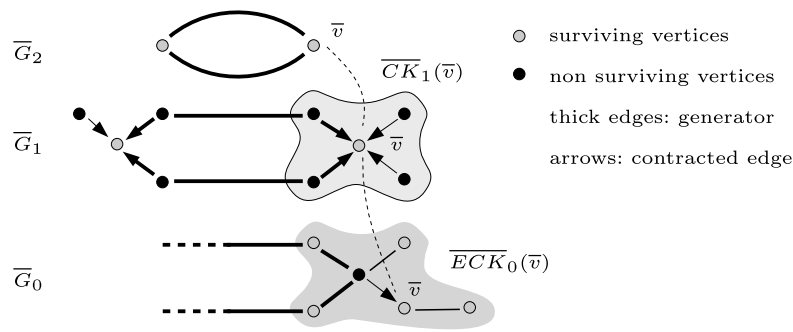


Fig. 4. Top-down delineation of a generator computed in \bar{G}_2 .

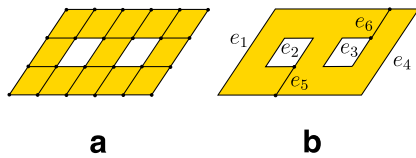


Fig. 5. (a) Bottom level, and (b) top level of the pyramid.

example $e_1 + e_4 + e_5$ is not a cycle. We need another edge to fulfill the cycle, and in Fig. 5(b), the possible one is the edge $-e_5$, so $e_1 + e_4 + e_5 - e_5 = e_1 + e_4$ is the cycle without the pseudo edge e_5 .

Property 2. The delineation of a top level 1-cycle that lies only on borders results in a 1-cycle in the bottom level that lies only on borders.

Proof. A 1D generator can be made out of more than one closed path, in this case its down-projection is done separately for each of them. The process of generator delineation (down-projection) presented in Section 4.3 requires for each of its closed paths:

- identifying in the bottom level the surviving edges that correspond to the given top level edges;
- adding paths connecting two identified edges, if their associated edges from the top level share a vertex (if two identified edges share a common vertex, no path is added).

The identified surviving edges are guaranteed to lie on borders because of their one to one association to their corresponding top level edges.

As presented in Section 4.3, each path added reconnects two consecutive surviving edges, and is a sub-path of the equivalent contraction kernel of the common vertex the two surviving edges share in the top level. Because the equivalent contraction kernels are trees, the added paths are unique [22].

Moreover, these paths lie on borders because:

- in the bottom level, for any two vertices of one border there are exactly two paths that connect them and which are made only of border edges;
- border edges are never removed (just contracted or surviving);

we can conclude that the unique path used to reconnect the vertices of two consecutive surviving border edges is made only of border edges. \square

5. Optimizations on the top level – the minimal generator set

We have shown in the previous section that our method provides a set of generators that always fit on the borders of the ob-

ject. As can be seen in Section 6, the set of generators that fit on the borders of an object is not uniquely defined. In this section, we go one step further and show that any of these possible sets of generators can be transformed such that each generator surrounds exactly one hole; we call this set the *minimal set of generators*.

Fig. 6 illustrates an object with 3 holes. Let a denote the outer border of the object, and let b, c and d denote the three closed curves corresponding to the borders of the three holes. Among others, $\{a, b, c\}$ and $\{b + c, c + d, d\}$ are possible basis of H_1 ; and $\{b, c, d\}$ is the minimal one, as each generator surrounds exactly one hole.

A set of generators is called *minimal* if none of the generators intersect, each surrounds exactly one hole, and all generators fit on borders. A minimal set of generators does not contain the outer border of the object (a in Fig. 6).

Definition 1. Two H_1 generators are *separate* if they neither intersect nor are in an inclusion relation.

Their interiors are disjoint. Note that the term *separate* has to be extended to higher dimensions as additional relations exist, e.g. in 3D, imagine two closed curves (rings), like the rings of a chain. By the previous definition they would be *separate* even though they are not.

Property 3. If all H_1 generators of a connected object in 2D are separate, each one surrounds one hole.

Proof. All the generators of a connected object in 2D are located within the object. Thus each H_1 will surround a hole of the 2D object. Since all the H_1 are separate, based on the definition, these generators do not intersect and their interior are disjoint. Therefore all these generators are distinct. \square

Let us call this set of generators *independent*.

Property 4. An independent set of generators fitting to the borders of the object is minimal.

Proof. The set of borders of an object consists of the k borders separating the object from its k holes (b, c and d in Fig. 6), and the outside border (a in Fig. 6), separating the object from the background. If the given set of generators is independent, the outside border cannot be one of them, and there can be no intersection or overlapping. The given generator set is made of the k borders surrounding the k holes – which make up the minimal generator set. \square

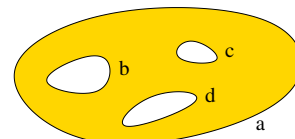


Fig. 6. A 2D shape with 3 holes.

Having an arbitrary set of generators it can be transformed to an independent set by the following operations:

- (1) **shrinking:** independently from all the others, each generator cycle is continuously transformed inside the object such that its interior is reduced to 0, with holes blocking the movement of the generator. Imagine a balloon with some solid objects inside (representing the holes) from which we draw out all the air;
- (2) **operating with generators:** if $\{a_1, \dots, a_n\}$ is a basis of a group, replacing any element a_i by $a_i + ka_j$, with $i \neq j$ and k an integer, will generate a basis of the same group. Any basis can be build from the minimal one. Looking at it the other way around, consider a basis containing an element a that surrounds k holes h_1, \dots, h_k , $k > 1$ then there is another element a' in this basis that surrounds l holes h'_1, \dots, h'_l with $k > l \geq 1$ and $h'_1, \dots, h'_l \subset h_1, \dots, h_k$. By replacing the element of the basis a by $a - a'$, we still have a basis, but now the new element $a - a'$ surrounds only $k - l < k$ holes. We can conclude that starting from any basis, we can always come up with a new basis where each element surrounds only one hole. Moreover, if we start with a basis where all the generators fit on borders, we obtain the minimal set. (Addition and subtraction of shrunken generators can be done by adding new edges and removing common ones.)

Table 2
The number of cells on the initial image and on the top of the pyramid

	Initial image			Top of the pyramid		
	0D-Cells	1D-Cells	2D-Cells	0D-Cells	1D-Cells	2D-Cells
Fig. 7	6036	11541	5503	7	10	1
Fig. 8	4905	9233	4325	11	17	3

Note that for the 2D case, computing the minimal generator set can be easily done by taking the borders separating the object from the holes. In the top level of the pyramid these holes are surrounded by self loops or by cycles with a reduced number of edges. If using concepts like in [23], enumerating all these edges is done in linear time in the number of border edges of the object in the reduced map. Extension to any dimension is planned.

6. Experiments on 2D images

We present and discuss initial experiments that have been performed on 2D images. For each shape, we have computed homology generators directly on the initial image, and on the top level of the pyramid (with and without Agoston). Note that for visualisation purposes each edge of a 1D generator is shown by pixels incident to it.

Table 2 shows the number of 0D, 1D and 2D-cells on the initial image and on the top level of the pyramid, for the shape presented in Figs. 7 and 8. One can observe that for each shape the total number of cells is considerably reduced on the highest level of the pyramid. Thus, the computation of homology generators can be done on much smaller matrices on the top level instead of the initial image. In Figs. 7 and 8, it can be seen that our new method provides a valid set of generators in each case.

Moreover, using the classical method, we cannot have any control of the geometry of the generators computed. More precisely, the aspect of the obtained generators is directly linked to the construction of incidence matrices, which is determined by the scanning of each cell of the initial image. The shape shown on Fig. 9 has been obtained by rotating Fig. 7. In Fig. 9(a), one can observe that the aspect of the generators computed on the initial image “follows” the scanning of the cells (from top to bottom, and left to right). The generators obtained in Fig. 9(b) always fit on the boundaries of the image (see Section 4.5).

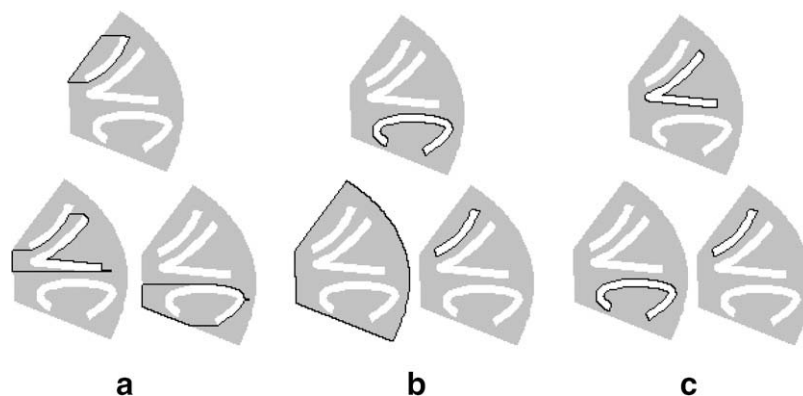


Fig. 7. (a) The homology generators computed on the initial image. (b) The down-projected generators. (c) Minimal set.

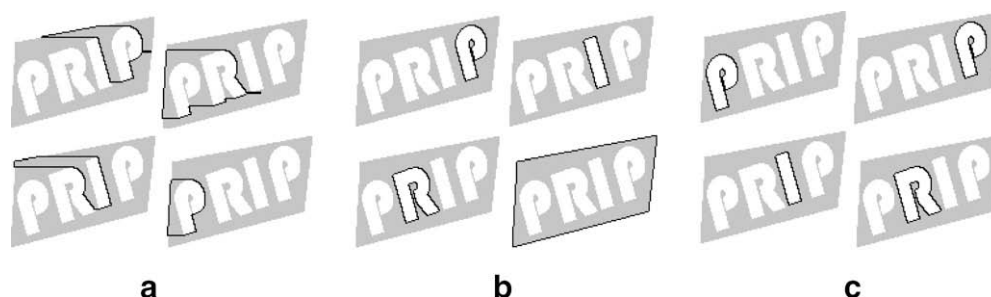


Fig. 8. (a) The homology generators computed on the initial image. (b) The down-projected generator. (c) Minimal set.

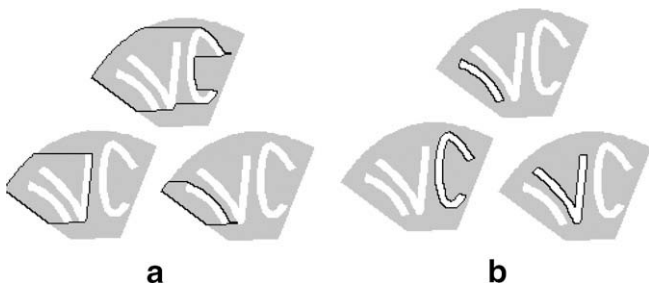


Fig. 9. Influence of the scanning (compare with Fig. 7 (a) and (b)).

One can note that the sets of cycles obtained in Fig. 7(a) and (b) do not surround the same (set of) 1D-holes of the shape S . Indeed, these two sets are two different basis of the same group $H_1(S)$: let i , v and c denote the equivalence class of cycles that surround respectively the letters 'I', 'V', and 'C'. The set of generators in Fig. 7(a) describe $H_1(S)$ in the basis $\{i, v, c\}$ whereas in Fig. 7(b), $H_1(S)$ is described in the basis $\{c, i + v + c, i\}$.

In Figs. 10 and 11 real world images are shown. We have first segmented the images (e.g. one can choose the minimum spanning tree based pyramid segmentation [24], and build generators on these segmented images – for clarity of the presentation we used a binary segmentation, Figs. 10(b) and 11(a)). Fig. 10(a) shows the original image, Fig. 10(b) the used binary segmentation, and Fig. 10(c) the brightened image with the obtained generators in black.

In Fig. 11 white means 1-dimensional hole. The basis in Fig. 11(b)–(d) are different but they are basis of the same first homology group. The generators shown in Figs. 11(c) and (d) are nicely fitted on the borders of regions (1D-holes).

7. Conclusion

We have presented a new method for computing homology groups of images and their generators, using irregular graph pyramids. The homology generators are computed efficiently on the top level of the pyramid, since the number of cells is small, and a top down process (down-projection) delineates the homology generators of the initial image. Some preliminary results have been

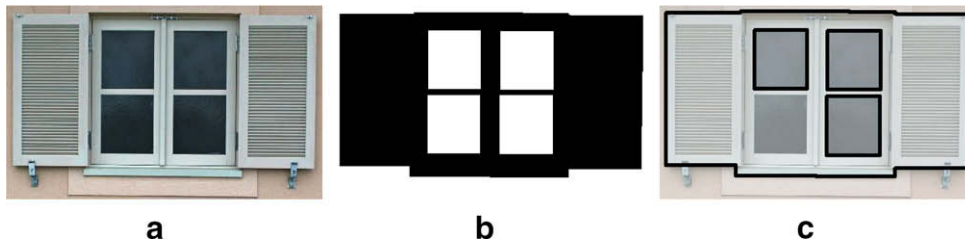


Fig. 10. (a) Original image. (b) Segmentation. (c) Down-projected generators (in black).

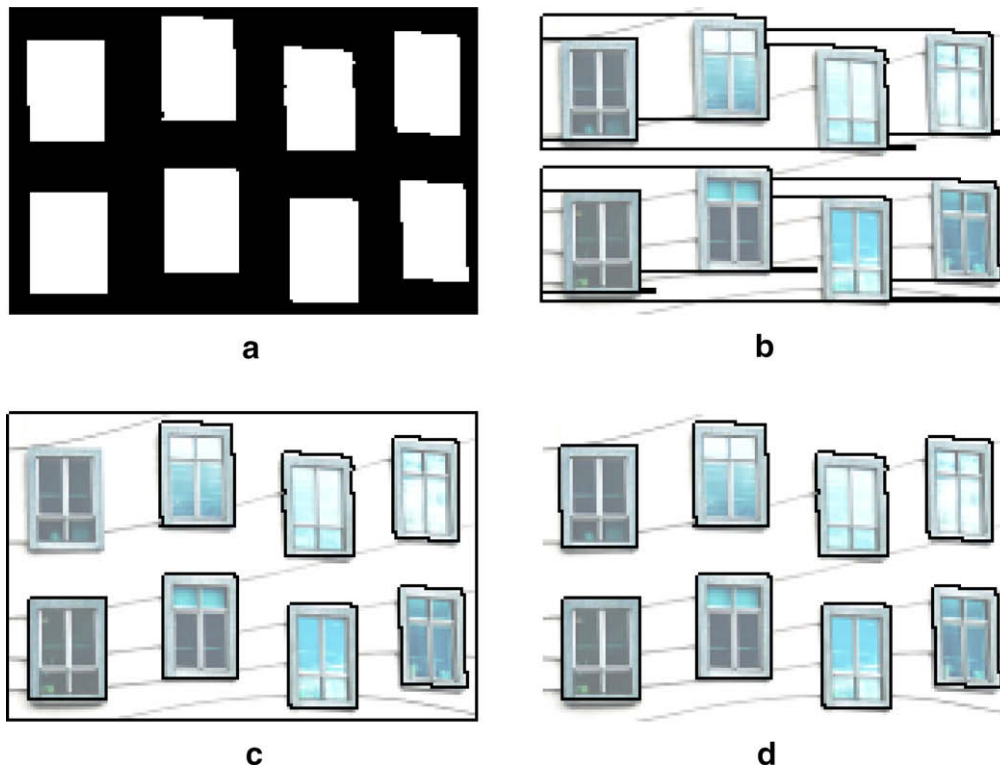


Fig. 11. (a) Segmented image. Generators overlaid on the original image. (b) The homology generators computed on the initial image. (c) The down-projected generator. (d) Minimal set.

shown for 2D binary images. The generators computed with this new method fit on the boundaries of objects. The concept of minimal generator set is defined and studied.

In a future work, we plan to extend this method to 3D and nD images, using the (already existing) structures of 3D and nD irregular pyramids. We also plan to use the minimal generator set and the property that down-projected generators always fit on boundaries, and apply homology generators for object matching and object tracking.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.imavis.2008.06.009.

References

- [1] V.A. Kovalevsky, Finite topology as applied to image analysis, *Computer Vision, Graphics, and Image Processing* 46 (1989) 141–161.
- [2] V.A. Kovalevsky, Digital geometry based on the topology of abstract cellular complexes, in: J.-M. Chassery, J. Francon, A. Montanvert, J.-P. Réveillès (Eds.), *Géométrie Discrète en Imagery, Fondements et Applications*, Strasbourg, France, 1993, pp. 259–284.
- [3] M.K. Agoston, Algebraic topology, a first course, in: Marcel Dekker (Ed.), *Pure and Applied Mathematics*, 1976.
- [4] M. Allili, K. Mischaikow, A. Tannenbaum, Cubical homology and the topological classification of 2d and 3d imagery, in: *Proceedings of International Conference Image Processing*, vol. 2, 2001, pp. 173–176.
- [5] R. Gonzales-Diaz, M. Jim'enez, M. Belén, P. Real, Extending the notion of at-model for integer homology, in: *Proceedings of the 15th International Workshop on Graph-based Representation for Pattern Recognition*, vol. 4538, No. 3, 2007, pp. 330–339.
- [6] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Brooks/Cole Publishing Company, 1999.
- [7] T. Kaczynski, K. Mischaikow, M. Mrozek, *Computational Homology*, Springer, 2004.
- [8] M. Niethammer, A.N. Stein, W.D. Kalies, P. Pilarczyk, K. Mischaikow, A. Tannenbaum, Analysis of blood vessels topology by cubical homology, in: *Proceedings of International Conference Image Processing*, vol. 2, 2002, pp. 969–972.
- [9] S. Peltier, A. Ion, Y. Haxhimusa, W. Kropatsch, G. Damiand, Computing homology group generators of images using irregular graph pyramids, in: *Proceedings of 6th IAPR International Workshop on Graph Based Representation in Pattern Recognition*, vol. 4538, Lecture Notes in Computer Science, Alicante, Spain, 2007, pp. 283–294.
- [10] G. Damiand, S. Peltier, L. Fuchs, Computing homology for surfaces with generalized maps: application to 3d images, in: *Proceedings of 2nd International Symposium on Visual Computing*, Lecture Notes in Computer Science, vol. 4292, Springer-Verlag, Lake Tahoe, Nevada, USA, 2006, pp. 1151–1160.
- [11] J.R. Munkres, *Elements of Algebraic Topology*, Perseus Books, 1984.
- [12] J.-M. Jolion, A. Rosenfeld, *A Pyramid Framework for Early Vision*, Kluwer, 1994.
- [13] P. Meer, Stochastic image pyramids, *Computer Vision, Graphics, and Image Processing* 45 (3) (1989) 269–294. also as UM CS TR-1871, June, 1987.
- [14] W.G. Kropatsch, Building irregular pyramids by dual graph contraction, *IEEE Proceedings Vision, Image and Signal Processing* 142 (6) (1995) 366–374.
- [15] R. Kannan, A. Bachem, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix, *SIAM Journal on Computing* 8 (4) (1979) 499–507.
- [16] J.-G. Dumas, F. Heckenbach, B.D. Saunders, V. Welker, Computing simplicial homology based on efficient smith normal form algorithms, in: *Algebra, Geometry, and Software Systems*, 2003, pp. 177–206.
- [17] A. Storjohann, Near optimal algorithms for computing smith normal forms of integer matrices, in: Y.N. Lakshman (Ed.), *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, ACM, 1996, pp. 267–274.
- [18] S. Peltier, S. Alayrangues, L. Fuchs, J.-O. Lachaud, Computation of homology groups and generators, *Computers and Graphics* 30 (2006) 62–69.
- [19] T. Kaczynski, M. Mrozek, M. Slusarek, Homology computation by reduction of chain complexes, *Computers & Mathematics with Applications* 34 (4) (1998) 59–70.
- [20] T. Kaczynski, K. Mischaikow, M. Mrozek, *Computational Homology*, Springer, 2004.
- [21] G. Damiand, P. Peltier, L. Fuchs, P. Lienhardt, Topological map: an efficient tool to compute incrementally topological features on 3d images, in: *Proceedings of 11th International Workshop on Combinatorial Image Analysis*, vol. 4040, 2006, pp. 1–15.
- [22] K. Thulasiraman, M.N.S. Swamy, *Graphs: Theory and Algorithms*, Wiley-Interscience, 1992.
- [23] L. Brun, W.G. Kropatsch, Contains and inside relationships within combinatorial pyramids, *Pattern Recognition* 39 (4) (2006) 515–526.
- [24] Y. Haxhimusa, W.G. Kropatsch, Hierarchy of partitions with dual graph contraction, in: B. Milaelis, G. Krell (Eds.), *Proceedings of German Pattern Recognition Symposium*, Lecture Notes in Computer Science, vol. 2781, Springer, Germany, 2003, pp. 338–345.