Controlling Geometry of Homology Generators

Abstract *Homology groups and their generators of a 2D image are computed using a hierarchical structure i.e. irregular graph pyramid. In this paper we show that the generators of the first homology groups of a 2D image, computed with this pyramid based method always fit on the boundaries of the regions.*

1 Introduction

A region/object 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, i.e. a partition of the object into cells of dimensions 0, 1, 2, 3 ... (i.e. vertices, edges, faces, volumes ...) [13]. In general, combinatorial structures (graphs, combinatorial maps, nG-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... (see [1]).

Homology is a powerful topological invariant, which characterizes an object by its "$p-$dimensional holes". Intuitively $0-$dimensional holes can be seen as connected components, $1-$dimensional holes can be seen as tunnels and $2-$dimensional holes as cavities. Unfortunately, there are no English notions for higher dimensional holes. This notion of $p-$dimensional hole is defined in any dimension. In Fig.1(a) an example of the torus is shown, which 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). Plainly, homology is a tool to study digital spaces, and has been applied for 2D and 3D image analysis [2]. Usage of homology groups and generators is a new topic and has been recently used in image processing. Although in this paper we use 2D images to show some nice properties of using homology groups and their generators in studying images, we do not encourage usage of homology groups and generators to find connected components in 2D image, since efficient approaches already exist [19]. 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 [9] to analyze 3D spatial structure of metals in an alloy and in medical image processing [17] in analyzing blood vessels. In higher dimensional problems (e.g. beating heart represented in 4D) the importance of homology groups and generators becomes clear in analyzing objects (their number of connected components, tunnels, holes, etc) in these spaces, because of the nice and clean formulations which hold in any dimension.

Moreover, if Betti numbers (rank of homology groups) provide the number of "p-dimensional" holes, a set of generators allows to locate them. In [18], it is shown that different parameters influence the geometry of the generators i.e. a generator can surround a "p-hole" more or less closely. A new method for computing homology groups and their generators is introduced in [5]. It uses a hierarchical structure based on a graph pyramid which is build by using two operations: contraction and removal. The main goal of this paper is to show that the generators build by the method in [5] is on the boundaries of the regions. We show this property by experimenting using 2D images and conjecture that this properties will hold also for higher dimensional data.

The paper is structured as follows. Basic notions of homology and irregular graph pyramids are recalled in Section 2 and 3. The new method to compute homology groups and their generators is presented in detail in Section 4. We finally show some experimental results on 2D images 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 [16].

The homology of a subdivided object X can be defined in an algebraic way by studying incidence relations of its

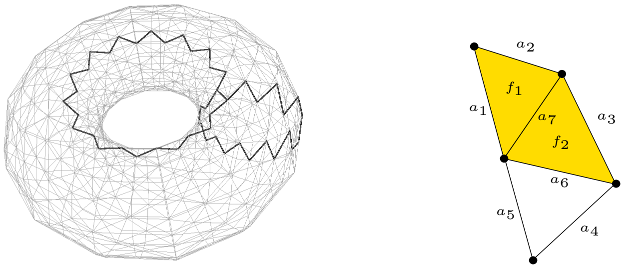


Figure 1: (a) : a triangulation of the torus. (b) : a simplicial complex made of 1 connected component and containing one $1-$dimensional hole.

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 $\sum_{i=1}^{nb\ p-cells} \alpha_i c_i$, 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. Anyway, the theorem of universal coefficients [16] ensures that all homological information is obtain by choosing $\mathfrak{A} = \mathbb{Z}$. It is also known [16] that for nD objects embedded in $\mathbb{R}^D$ the homology information can be computed considering simply chains with moduli 2 coefficients ($\mathfrak{A} = \mathbb{Z}/2\mathbb{Z}$). Note that is this case, a cell that appears twice on a chain vanishes, because $c + c = 0$ for any cell c when using moduli 2 coefficients. On 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 $p = 0, \ldots, n$, 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:

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

which satisfy $\partial_p \partial_{p-1}(c) = 0$ for any $p-$chain c, $p = 1..n$. 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, any cell that appears twice vanishes.

For each dimension $p = 0 \ldots n$, 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 $a_7 + a_5 + a_4 + a_3$ are $1-$cycles. The set of p-chains which bound 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, for $p = 0 \ldots n$, 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 $z_3 = a_1 + a_2 + a_3$; 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 an homology group generated by q independent equivalence classes $\mathcal{C}_1, \cdots, \mathcal{C}_q$, any set $\{h_1, \cdots, h_q \mid h_1 \in \mathcal{C}_1, \cdots, h_q \in \mathcal{C}_q\}$ is called a *set of generators* for H_p. For example, either $\{z_1\}, \{z_2\}$ can be chosen as a generator of H_1 for the object represented in Fig.1(b).

Note that some notions mentioned above can be confusing with similar notions in the graph theory field. Tab.1 associates these homology with notions classically used in graph theory.

3 Irregular Graph Pyramids

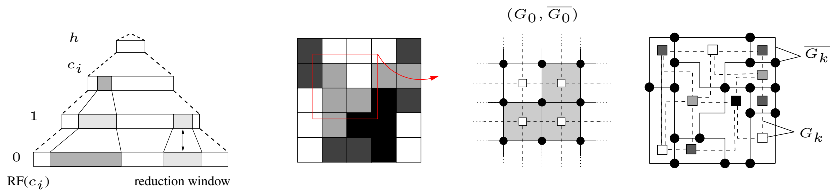
In this part, basic notions of pyramids like receptive field, contraction kernel, and equivalent contraction kernel are introduced, for more details see [8].

A pyramid (Fig. 2(a) 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 *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 is computed by means of a *reduction function* the input of which are the descriptions of the cells in the reduction window. 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 [15]. On the base level (level 0) of an irregular pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4(8)-connectivity of the pixels. A cell on level $k + 1$ (parent) is a union of some neighboring cells on level k (chil-

Table 1: Translation of homology notions in the graph field.

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



a) Discrete levels b) Image to dual graphs

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

dren). This union is controlled by so called *contraction kernels* (CK) [14], a spanning forest which relate two successive level 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 description 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. 2(b). 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. 2(b). The edges of $\overline{G_k}$ represent the borders of the cells on level k, solid lines in Fig. 2(b), 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. 2(b)), 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

There exists a general method for computing homology groups. This method is based on the transformation of incidence matrices [16] (i.e. which describe the boundary homomorphisms) into their reduced form called *Smith normal form*. Agoston proposes a general algorithm, based on the use of slightly modified Smith normal form, for computing a set of generators of these groups [1]. 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 [12]. 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 [4] and [20], an optimisation for the computation of homology generators, based on the use of sparse matrices and moduli operations has been proposed [18]. In particular, 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 have been proposed in [11]. The main idea of this algorithm is to reduce the number of cells of an initial object in order to obtain an homologically equivalent object, made 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 this work, an algorithm for computing a minimal representation of the boundary of a 3D voxel region, from which homology generators can directly be deduced have been defined in [3].

4.1 Generator Computation using Pyramids (GCP)

The GCP method proposed in [5] follows the same idea as the methods of Kaczynski and Damiand [10, 6]: 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 the pyramid. In this way, homology generators can be computed at the top level of the pyramid, and can be used to deduce generators of any lower level of the pyramid. The generators of the higher level can be directly down-projected on the desired level (using equivalent contraction kernels).

Starting from an initial image, an irregular graph pyramid is build. This method is valid as long as the algorithm used for the construction of the pyramid preserves homology. In particular, it is shown in [5] that the decimation by contraction kernels, described in section 3, preserves homology. Indeed, homology of the initial image can thus be computed in any level of the pyramid, and in particular on the top level where the object is described with the smallest number of cells.

The GCP method is summarized into the following steps:

1 Starting from labeled image, a graph pyramid $\{G_0, G_1, \ldots, G_k\}$ is built using contraction kernels of cells with the same label.

2 Homology groups generators are computed for G_k, using Agoston's method.

3 Homology generators of any level i can be deduced from those of level $i+1$ using the contraction kernels. In particular, we obtain the homology generators of the initial image.

Fig.3 illustrates the general method that we propose for computing homology generators of an image.

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 GCP method provides a set of generators that always fit on some boundaries of a region R. In the following, an edge on the boundary of a region is called a boundary edge.

First, we show that any $1-$cycle in the top level of the pyramid computed with GCP method contains only boundary edges. Second, we show that the down-projection of a $1-$cycle composed of boundary edges, is still a cycle composed of boundary edges.

Property 1 *Any $1-$cycle in the top level of the pyramid computed with GCP method contains only boundary 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 boundary edge or links two boundaries 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 boundaries of R, then R is made of 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 boundary edges.
□

Let us consider Fig. 4(b), which represents the top level of the pyramid built from the initial image represented in Fig. 4(a). The subdivision is made of one 2D-cell R_1; four boundary edges e_1, e_2, e_3, e_4; two pseudo edges e_5, e_6; and four vertices. The 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 $\alpha_i = 0, 1, i = 1 \ldots 4$.

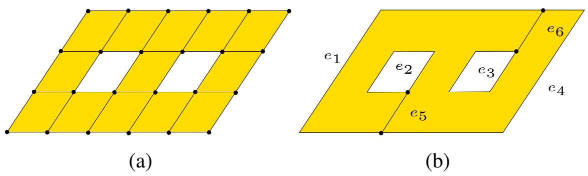


Figure 4: (a) Bottom level, and (b) top level of the pyramid.

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

Proof:
The process of generator delineation (down-projection) presented in [5] requires identifying in the bottom level the surviving edges that correspond to the given top level edges and where the generator cycles are disconnected, adding paths to reconnect.

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

As presented in [5], 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

- for any two vertices in any tree, there is a unique path connecting them [21],
- for any two vertices on the boundary (disconnected end-vertices of the two surviving edges) there are two paths that connect them and which are made only of boundary edges, and
- boundary edges are never removed [5] (just contracted or surviving),

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

6 Experiments on 2D Images

We present and discuss some experiments that have been performed on 2D images. We compute homology generators, for each region in two different ways: directly on the initial image (bottom level), and on the top level of the pyramid build on this image.

One can note that the set of cycles obtained in Fig.5(a) and Fig.5(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 a, b and c denote the equivalence class of cycles that surround respectively the left eye, the right eye, and the mouth. The set of generators in Fig.5(a)

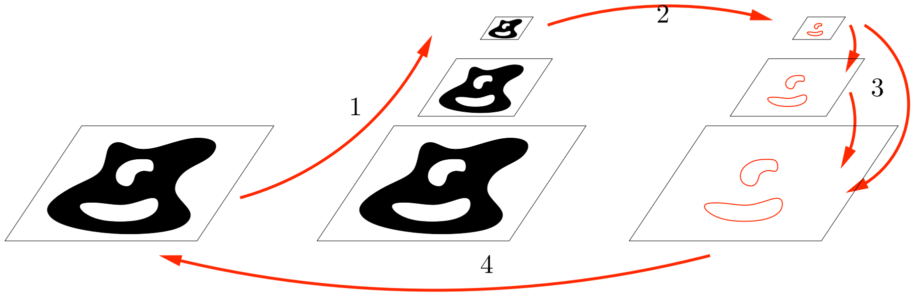


Figure 3: Computing generators of homology groups using an image pyramid.

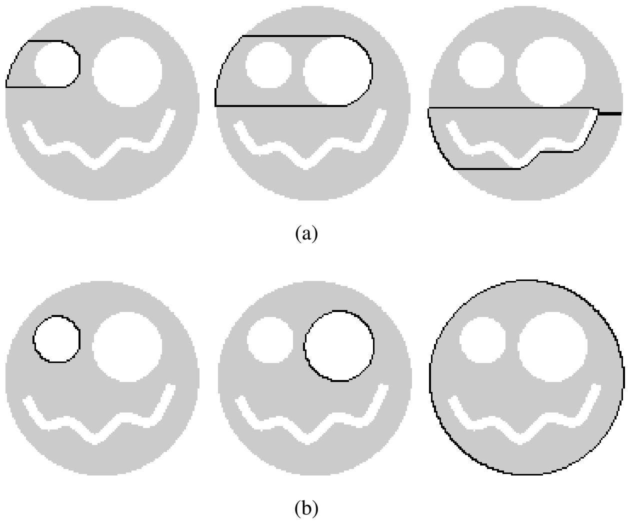


Figure 5: Generators overlayed on the image (a): the homology generators computed on the initial image, (b): GCP generators.

describe $H_1(S)$ in the basis $\{a+b, c, a\}$ whereas in Fig.5(b), $H_1(S)$ is described in the basis $\{a, a+b+c, b\}$. Note that in this figure we have put one generators (shown in black) per image.

In Fig.6 and Fig.7 some real world images are shown. We have first segmented the images (e.g. one can choose the minimum spanning tree based pyramid segmentation [7]). In principle one can build generators on these segmented images, but for clarity of this presentation we used binary segmentation (Fig.6(a) and 7(a)). In these binary images white means $1-$dimensional hole. Note that for visualization purposes we show with the gray color an island in Fig.6(a) that is not a $1-$dimensional hole since it is not enclosed by the black region. The basis in Fig.6(b) and in Fig.6(c) are different but they are basis of the same first homology group. The same holds for images Fig.7(b) and Fig.7(c).

The GCP generators shown in Fig.6(c) and Fig.7(b) are nicely fitted on the boundaries of regions ($1D-$holes). Note that the generators in Fig.6(b),6(c) and Fig.5(a),5(b) are shown with red and overlayed on the original image.

7 Conclusion

The GCP method for computing homology groups and their generators of images, using irregular graph pyramids has the nice property that the build generators always fit on the boundaries of the regions in 2D images. 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 results have been shown for 2D binary images.

In future work, we plan to study geometrical properties of homology generators computed with the GCP method for 3D images. In particular, we expect similar properties for homology generators of dimensions 1 and 2 (i.e. tunnels and cavities). We also plan to use these 'geometrically controlled' generators for object matching.

References

[1] M. K. Agoston. *Algebraic Topology, a first course*. Pure and applied mathematics. Marcel Dekker Ed., 1976.

[2] M. Allili, K. Mischaikow, and A. Tannenbaum. Cubical homology and the topological classification of 2d and 3d imagery. In *Proceedings of International Conference Image Processing*, volume 2, pages 173–176, 2001.

[3] G. Damiand, S. Peltier, and L. Fuchs. Computing homology for surfaces with generalized maps: Application to 3d images. In *Procedings of 2nd International Symposium on Visual Computing*, volume 4292 of *LNCS*, pages 1151–1160, Lake Tahoe, Nevada, USA, November 2006. Springer-verlag.

[4] J.-G. Dumas, F. Heckenbach, B. D. Saunders, and V Welker. Computing simplicial homology based on efficient smith normal form algorithms. In *Algebra, Geometry, and Software Systems*, pages 177–206, 2003.

[5] Removed for blind reviewing. Computing homology group generators of images using irregular graph pyramids. Technical report, 2007.

[6] Damiand G., Peltier P., Fuchs L., and Lienhardt P. Topological map: An efficient tool to compute incrementally topological features on 3d images. In *Proceedings of 11th International Workshop on Combinatorial Image Analysis*, volume 4040, pages 1–15, June 2006.

[7] Yll Haxhimusa and Walter G. Kropatsch. Hierarchy of partitions with dual graph contraction. In B. Milaelis and G. Krell, editors, *Proceedings of German Pattern Recognition Symposium*, volume 2781 of *Lecture Notes in Computer Science*, pages 338–345, Germany, 2003. Springer.

[8] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision.* Kluwer, 1994.

[9] T. Kaczynksi, K. Mischaikow, and M. Mrozek. *Computational Homology.* Springer, 2004.

[10] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Springer, 2004.

[11] T. Kaczynski, M. Mrozek, and M. Slusarek. Homology computation by reduction of chain complexes. *Computers & Math. Appl.*, 34(4):59–70, 1998.

[12] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, November 1979.

[13] Vladimir A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46:141–161, 1989.

[14] Walter G. Kropatsch. Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374, December 1995.

[15] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, March 1989. Also as UM CS TR-1871, June, 1987.

[16] J. R. Munkres. *Elements of algebraic topology*. Perseus Books, 1984.

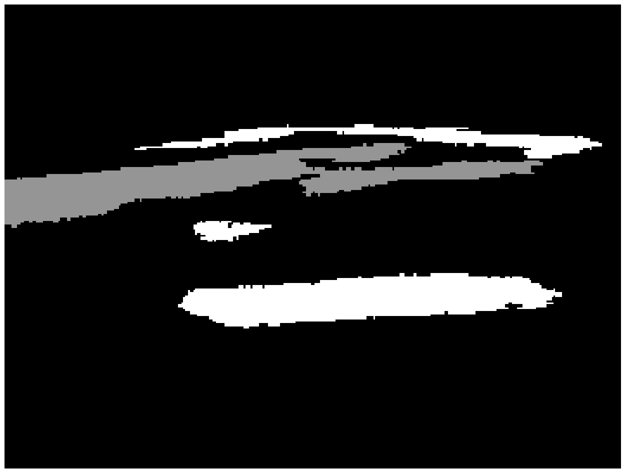
[17] M. Niethammer, A. N. Stein, W. D. Kalies, P. Pilarczyk, K. Mischaikow, and A. Tannenbaum. Analysis of blood vessels topology by cubical homology. In *Proceedings of International Conference Image Processing*, volume 2, pages 969–972, 2002.

[18] S. Peltier, S. Alayrangues, L. Fuchs, and J.-O. Lachaud. Computation of homology groups and generators. *Computers and graphics*, 30:62–69, febuary 2006.

[19] Milan Sonka, Vlaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company, 1999.

[20] A. Storjohann. Near optimal algorithms for computing smith normal forms of integer matrices. In Y. N. Lakshman, editor, *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 267–274. ACM, 1996.

[21] K. Thulasiraman and M. N. S. Swamy. *Graphs: Theory and Algorithms.* Wiley-Interscience, 1992.

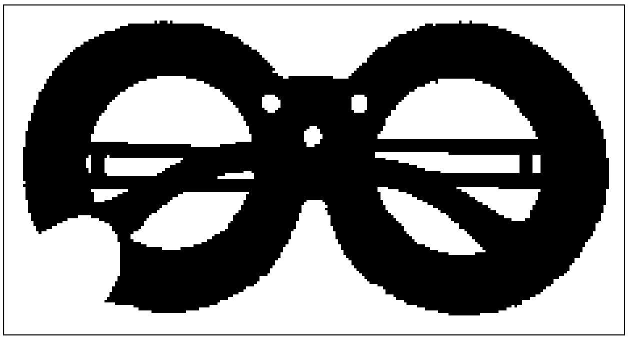


(a)

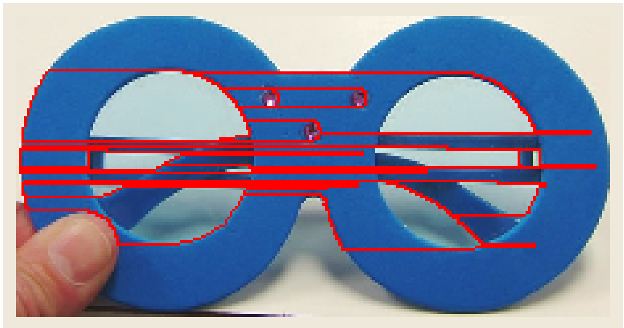
(b)

(c)

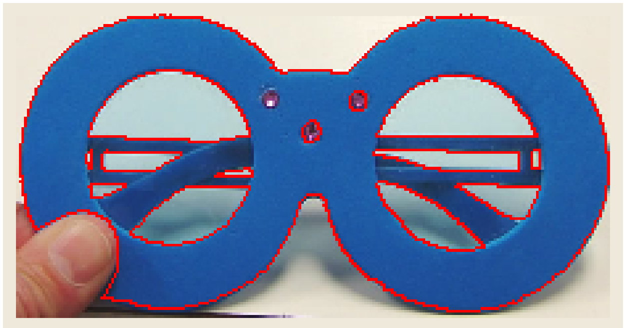
Figure 6: (a): segmented image. Generators overlayed on the original image (b): the homology generators computed on the initial image, (c): GCP generators.



(a)



(b)



(c)

Figure 7: (a): segmented image. Generators overlayed on the original image (b): the homology generators computed on the initial image, (c): GCP generators.

Document is too long. Maximum of 6 pages is allowed! In

Document is too long. Maximum of 6 pages is allowed! In